WMT 2022

Seventh Conference on Machine Translation

Proceedings of the Conference

December 7-8, 2022

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-959429-29-6

Introduction

The Seventh Conference on Machine Translation (WMT 2022) took place on Wednesday, December 7 and Thursday, December 8, 2022 immediately preceding the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2022).

This is the seventh time WMT has been held as a conference. The first time WMT was held as a conference was at ACL 2016 in Berlin, Germany, the second time at EMNLP 2017 in Copenhagen, Denmark, the third time at EMNLP 2028 in Brussels, Belgium, the fourth time at ACL 2019 in Florence, Italy, the fifth time at EMNLP-2020, which was held as an online event due to the COVID-19 pandemic, and the sixth time at EMNLP 2021 at Punta Cana, Dominican Republic. Prior to being a conference, WMT was held 10 times as a workshop. WMT was held for the first time at HLT-NAACL 2006 in New York City, USA. In the following years the Workshop on Statistical Machine Translation was held at ACL 2007 in Prague, Czech Republic, ACL 2008, Columbus, Ohio, USA, EACL 2009 in Athens, Greece, ACL 2010 in Uppsala, Sweden, EMNLP 2011 in Edinburgh, Scotland, NAACL 2012 in Montreal, Canada, ACL 2013 in Sofia, Bulgaria, ACL 2014 in Baltimore, USA, EMNLP 2015 in Lisbon, Portugal.

The focus of our conference is to bring together researchers from the area of machine translation and invite selected research papers to be presented at the conference.

Prior to the conference, in addition to soliciting relevant papers for review and possible presentation, we conducted 13 shared tasks: General Machine Translation of News, Similar Language Translation, Biomedical Translation, Large-Scale Machine Translation Evaluation for African Languages, Translation Efficiency, Sign Language Translation, Code-mixed Machine Translation, Chat Translation Task, Unsupervised MT and Very Low Resource Supervised Machine Translation, Metrics for Machine Translation, Quality Estimation of Translation, Word-Level Auto-Completion, Translation Suggestion, and the Automatic Post-Editing task.

The results of all shared tasks were announced at the conference, and these proceedings also include overview papers for the shared tasks, summarizing the results, as well as providing information about the data used and any procedures that were followed in conducting or scoring the tasks. In addition, there are short papers from each participating team that describe their underlying system in greater detail.

Like in previous years, we have received a far larger number of submissions than we could accept for presentation. WMT 2022 has received 48 full research paper submissions (not counting withdrawn submissions). In total, WMT 2022 featured 16 full research paper presentations and 96 shared task presentations.

The invited talk entitled was given by Ondřej Bojar from Charles University, Prague, Czech Republic

We would like to thank the members of the Program Committee for their timely reviews. We also would like to thank the participants of the shared task and all the other volunteers who helped with the evaluations.

Loïc Barrault, Rachel Bawden, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Anton Dvorkovich, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Rebecca Knowles, Tom Kocmi, Philipp Koehn, André Martins, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Mariya Shmatova, Marco Turchi, and Marcos Zampieri

Co-Organizers

Organizing Committee

Organizers

Loïc Barrault, Meta AI
Ondřej Bojar, Charles University
Fethi Bougares, University of Le Mans
Rajen Chatterjee, Apple
Marta R. Costa-jussà, Meta AI
Christian Federmann, Microsoft
Mark Fishel, University of Tartu
Alexander Fraser, LMU Munich
Markus Freitag, Google
Yvette Graham, Dublin City University
Roman Grundkiewicz, Microsoft
Paco Guzman, Meta AI
Barry Haddow, University of Edinburgh
Matthias Huck, LMU Munich
Antonio Jimeno Yepes, RMIT University
Tom Kocmi, Microsoft
Philipp Koehn, Johns Hopkins University
André Martins, Unbabel
Christof Monz, University of Amsterdam
Makoto Morishita, NTT
Masaaki Nagata, NTT
Toshiaki Nakazawa, University of Tokyo
Matteo Negri, FBK
Aurélie Névéol, LIMSI, CNRS
Mariana Neves, German Federal Institute for Risk Assessment
Martin Popel, Charles University
Matt Post, Johns Hopkins University
Mariya Shmatova, Neurodub
Marco Turchi, FBK
Marcos Zampieri, Rochester Institute of Technology

Program Committee

Reviewers

David Adelani, Jesujoba Alabi, Antonios Anastasopoulos, Philip Arthur, Duygu Ataman, Eleftherios Avramidis

Parnia Bahar, Antonio Valerio Miceli Barone, Maximiliana Behnke, Meriem Beloucif, Toms Bergmanis, Frédéric Blain, Laurie Burchell, Bill Byrne

Ozan Caglayan, Sheila Castilho, Pinzhen Chen, Colin Cherry, Vishal Chowdhary, Chenhui Chu, Ann Clifton, Marta R. Costa-jussà, Josep Crego

Raj Dabre, Steve DeNeefe, Michael Denkowski, Shuoyang Ding, Miguel Domingo, Kevin Duh

Hiroshi Echizen'ya, Carla Parra Escartín, Cristina España-Bonet, Miquel Esplà-Gomis

Natalia Fedorova, Yang Feng, Orhan Firat, Mikel L. Forcada, George Foster, Atsushi Fujita

Mercedes García-Martínez, Ekaterina Garmash, Jesús González-Rubio, Isao Goto, Cyril Goutte, Mandy Guo, Jeremy Gwinnup

Viktor Hangya, Greg Hanneman, Jindřich Helcl, John Henderson, Amr Hendy, Dam Heo, Nico Herbig, Christian Herold, Felix Hieber, Cong Duy Vu Hoang, Hieu Hoang, Yongkeun Hwang

Kenji Imamura

Josef Jon

Diptesh Kanojia, Hyunjoong Kim, Yunsu Kim, Elizaveta Korotkova, Julia Kreutzer, Mateusz Krubiński, Roland Kuhn, Gaurav Kumar, Shankar Kumar, Anoop Kunchukuttan

Samuel Läubli, Wen Lai, Surafel M. Lakew, Ekaterina Lapshinova-Koltunski, Samuel Larkin, Giang Le, WonKee Lee, Els Lefever, Gregor Leusch, Bei Li, Zhenhao Li, Jindřich Libovický, Seunghyun Lim, Patrick Littell, Danni Liu, Qun Liu

Mathias Müller, Vivien Macketanz, Jean Maillard, Andreas Maletti, Marion Weller-Di Marco, Vukosi Marivate, Arne Mauser, Arya D. McCarthy, Kenton Murray, Tomáš Musil

Ajay Nagesh, Graeme Nail, Graham Neubig, Jan Niehues, Vassilina Nikoulina, Xing Niu, Michal Novák

Tsuyoshi Okita

Santanu Pal, Pavel Pecina, Stephan Peitz, Sergio Penkale, Mārcis Pinnis, Maja Popović

Sadaf Abdul Rauf, Vikas Raunak, Matīss Rikters, Annette Rios

Víctor M. Sánchez-Cartagena, Felipe Sánchez-Martínez, Safiyyah Saleem, Elizabeth Salesky, Hassan Sawaf, Carolina Scarton, Holger Schwenk, Rico Sennrich, Patrick Simianer, Felix Stahlberg, David Stap, Katsuhito Sudoh

Aleš Tamchyna, Gongbo Tang, Brian Thompson, Jörg Tiedemann, Antonio Toral, Ke Tran, Ferhan Ture

Masao Utiyama

David Vilar, Ekaterina Vylomova

Wei Wang, Weiyue Wang, Taro Watanabe, Gideon Maillette de Buy Wenniger, Guillaume Wenzek, Hua Wu

Tong Xiao, Jinan Xu, Jitao Xu

Yinfei Yang, Hyeongu Yun, François Yvon

Xianfeng Zeng, Dakun Zhang, Zhong Zhou

Keynote Talk: Speech Translation — When Two Superhuman Technologies Combined Fail

Ondrej Bojar
Charles University

Abstract: In my talk, I will describe the challenges we faced in the EU project ELITR when we put together low-latency speech recognition and machine translation, both ’at or above human levels of performance’ in order to simultaneously translate spontaneous speech to 42 languages. I will have to talk about various aspects of quality and I will delve into both source and target multilinguality. Our ongoing quest is to put human interpreters on the same scales as machines, to contrast their virtues, complement them in their service and build upon their output.

Bio: Ondřej Bojar is an associate professor at ÚFAL, Charles University, and a lead scientist in Machine Translation in the Czech Republic. He has been co-organizing WMT shared tasks in machine translation and machine translation evaluation since 2013. His system has dominated English-Czech translation in the years 2013-2015, before deep learning and neural networks fundamentally changed the field. Having taken part and later supervised ÚFAL’s participation in a series of EU projects (EuroMatrix, EuroMatrixPlus, MosesCore, QT21, HimL, CRACKER, Bergamot), he has recently concluded his coordination of the EU project ELITR (http://elitr.eu/) focussed on simultaneous speech translation into over 40 languages. ELITR has also coined the task of project meeting summarization with its AutoMin 2021 shared task.

Table of Contents

Program

Wednesday, December 7, 2022

09:00 - 09:10 *Opening Remarks*

09:10 - 10:30 *Session 1 — Shared Task Overview Papers I*

Findings of the 2022 Conference on Machine Translation (WMT22)
Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel and Maja Popović

Results of WMT22 Metrics Shared Task: Stop Using BLEU – Neural Metrics Are Better and More Robust
Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie and André F. T. Martins

Findings of the WMT 2022 Shared Task on Quality Estimation
Chrysoula Zerva, Frédéric Blain, Ricardo Rei, Piyawat Lertvittayakumjorn, José G. C. De Souza, Steffen Eger, Diptesh Kanojia, Duarte Alves, Constantin Orăsan, Marina Fomicheva, André F. T. Martins and Lucia Specia

Findings of the WMT 2022 Shared Task on Efficient Translation
Kenneth Heafield, Biao Zhang, Graeme Nail, Jelmer Van Der Linde and Nikolay Bogoychev

Findings of the WMT 2022 Shared Task on Automatic Post-Editing
Pushpak Bhattacharyya, Rajen Chatterjee, Markus Freitag, Diptesh Kanojia, Matteo Negri and Marco Turchi

10:30 - 11:00 *Coffee Break*

11:00 - 12:40 *Session 2 — Research Papers on Evaluation and Bias*

Embarrassingly Easy Document-Level MT Metrics: How to Convert Any Pretrained Metric into a Document-Level Metric
Giorgos Vernikos, Brian Thompson, Prashant Mathur and Marcello Federico

Searching for a Higher Power in the Human Evaluation of MT
Johnny Wei, Tom Kocmi and Christian Federmann

Analyzing the Use of Influence Functions for Instance-Specific Data Filtering in Neural Machine Translation
Tsz Kin Lam, Eva Hasler and Felix Hieber

Wednesday, December 7, 2022 (continued)

Continuous Rating as Reliable Human Evaluation of Simultaneous Speech Translation
Dávid Javorský, Dominik Macháček and Ondřej Bojar

Gender Bias Mitigation for NMT Involving Genderless Languages
Ander Corral and Xabier Saralegi

12:40 - 14:00 *Lunch Break*

14:00 - 15:20 *Session 3 — Research Papers on Multilingual, Multimodal, Multidomain Translation*

Exploring the Benefits and Limitations of Multilinguality for Non-autoregressive Machine Translation
Sweta Agrawal, Julia Kreutzer and Colin Cherry

Learning an Artificial Language for Knowledge-Sharing in Multilingual Translation
Danni Liu and Jan Niehues

Don't Discard Fixed-Window Audio Segmentation in Speech-to-Text Translation
Chantal Amrhein and Barry Haddow

Additive Interventions Yield Robust Multi-Domain Machine Translation Models
Elijah Rippeth and Matt Post

Test Set Sampling Affects System Rankings: Expanded Human Evaluation of WMT20 English-Inuktitut Systems
Rebecca Knowles and Chi-kiu Lo

Can Domains Be Transferred across Languages in Multi-Domain Multilingual Neural Machine Translation?
Thuy-trang Vu, Shahram Khadivi, Xuanli He, Dinh Phung and Gholamreza Haffari

15:20 - 16:00 *Coffee Break*

16:00 - 17:30 *Session 4 — Shared Task Sytem Description Posters I*

Wednesday, December 7, 2022 (continued)

16:00 - 17:30 *News Translation Task*

Inria-ALMAnaCH at WMT 2022: Does Transcription Help Cross-Script Machine Translation?
Jesujoba Alabi, Lydia Nishimwe, Benjamin Muller, Camille Rey, Benoît Sagot and Rachel Bawden

NAIST-NICT-TIT WMT22 General MT Task Submission
Hiroyuki Deguchi, Kenji Imamura, Masahiro Kaneko, Yuto Nishida, Yusuke Sakai, Justin Vasselli, Huy Hien Vu and Taro Watanabe

Samsung R&D Institute Poland Participation in WMT 2022
Adam Dobrowolski, Mateusz Klimaszewski, Adam Myśliwy, Marcin Szymański, Jakub Kowalski, Kornelia Szypuła, Paweł Przewłocki and Paweł Przybysz

Tencent AI Lab - Shanghai Jiao Tong University Low-Resource Translation System for the WMT22 Translation Task
Zhiwei He, Xing Wang, Zhaopeng Tu, Shuming Shi and Rui Wang

Lan-Bridge MT's Participation in the WMT 2022 General Translation Shared Task
Bing Han, Yangjian Wu, Gang Hu and Qiulin Chen

Manifold's English-Chinese System at WMT22 General MT Task
Chang Jin, Tingxun Shi, Zhengshan Xue and Xiaodong Lin

CUNI-Bergamot Submission at WMT22 General Translation Task
Josef Jon, Martin Popel and Ondřej Bojar

KYB General Machine Translation Systems for WMT22
Shivam Kalkar, Yoko Matsuzaki and Ben Li

The AISP-SJTU Translation System for WMT 2022
Guangfeng Liu, Qinpei Zhu, Xingyu Chen, Renjie Feng, Jianxin Ren, Renshou Wu, Qingliang Miao, Rui Wang and Kai Yu

NT5 at WMT 2022 General Translation Task
Makoto Morishita, Keito Kudo, Yui Oka, Katsuki Chousa, Shun Kiyono, Sho Takase and Jun Suzuki

Wednesday, December 7, 2022 (continued)

Adam Mickiewicz University at WMT 2022: NER-Assisted and Quality-Aware Neural Machine Translation
Artur Nowakowski, Gabriela Pałka, Kamil Guttmann and Mikołaj Pokrywka

Evaluating Corpus Cleanup Methods in the WMT'22 News Translation Task
Marilena Malli and George Tambouratzis

PROMT Systems for WMT22 General Translation Task
Alexander Molchanov, Vladislav Kovalenko and Natalia Makhamalkina

eTranslation's Submissions to the WMT22 General Machine Translation Task
Csaba Oravecz, Katina Bontcheva, David Kolovratnìk, Bogomil Kovachev and Christopher Scott

CUNI Systems for the WMT 22 Czech-Ukrainian Translation Task
Martin Popel, Jindřich Libovický and Jindřich Helcl

The ARC-NKUA Submission for the English-Ukrainian General Machine Translation Shared Task at WMT22
Dimitrios Roussis and Vassilis Papavassiliou

The NiuTrans Machine Translation Systems for WMT22
Weiqiao Shan, Zhiquan Cao, Yuchen Han, Siming Wu, Yimin Hu, Jie Wang, Yi Zhang, Hou Baoyu, Hang Cao, Chenghao Gao, Xiaowen Liu, Tong Xiao, Anxiang Ma and Jingbo Zhu

Teaching Unseen Low-resource Languages to Large Translation Models
Maali Tars, Taido Purason and Andre Tättar

DUTNLP Machine Translation System for WMT22 General MT Task
Ting Wang, Huan Liu, Junpeng Liu and Degen Huang

HW-TSC's Submissions to the WMT 2022 General Machine Translation Shared Task
Daimeng Wei, Zhiqiang Rao, Zhanglin Wu, Shaojun Li, Yuanchang Luo, Yuhao Xie, Xiaoyu Chen, Hengchao Shang, Zongyao Li, Zhengzhe Yu, Jinlong Yang, Miaomiao Ma, Lizhi Lei, Hao Yang and Ying Qin

Vega-MT: The JD Explore Academy Machine Translation System for WMT22
Changtong Zan, Keqin Peng, Liang Ding, Baopu Qiu, Boan Liu, Shwai He, Qingyu Lu, Zheng Zhang, Chuang Liu, Weifeng Liu, Yibing Zhan and Dacheng Tao

No Domain Left behind
Hui Zeng

GTCOM Neural Machine Translation Systems for WMT22
Hao Zong and Chao Bei

16:00 - 17:30 *Test Suites*

Linguistically Motivated Evaluation of the 2022 State-of-the-art Machine Translation Systems for Three Language Directions
Vivien Macketanz, Shushen Manakhimova, Eleftherios Avramidis, Ekaterina Lapshinova-koltunski, Sergei Bagdasarov and Sebastian Möller

Automated Evaluation Metric for Terminology Consistency in MT
Kirill Semenov and Ondřej Bojar

Test Suite Evaluation: Morphological Challenges and Pronoun Translation
Marion Weller-di Marco and Alexander Fraser

16:00 - 17:30 *Metrics Task*

Robust MT Evaluation with Sentence-level Multilingual Augmentation
Duarte Alves, Ricardo Rei, Ana C Farinha, José G. C. De Souza and André F. T. Martins

ACES: Translation Accuracy Challenge Sets for Evaluating Machine Translation Metrics
Chantal Amrhein, Nikita Moghe and Liane Guillou

Linguistically Motivated Evaluation of Machine Translation Metrics Based on a Challenge Set
Eleftherios Avramidis and Vivien Macketanz

Exploring Robustness of Machine Translation Metrics: A Study of Twenty-Two Automatic Metrics in the WMT22 Metric Task
Xiaoyu Chen, Daimeng Wei, Hengchao Shang, Zongyao Li, Zhanglin Wu, Zhengzhe Yu, Ting Zhu, Mengli Zhu, Ning Xie, Lizhi Lei, Shimin Tao, Hao Yang and Ying Qin

MS-COMET: More and Better Human Judgements Improve Metric Performance
Tom Kocmi, Hitokazu Matsushita and Christian Federmann

Wednesday, December 7, 2022 (continued)

Partial Could Be Better than Whole. HW-TSC 2022 Submission for the Metrics Shared Task
Yilun Liu, Xiaosong Qiao, Zhanglin Wu, Su Chang, Min Zhang, Yanqing Zhao, Song Peng, Shimin Tao, Hao Yang, Ying Qin, Jiaxin Guo, Minghan Wang, Yinglu Li, Peng Li and Xiaofeng Zhao

Unsupervised Embedding-based Metric for MT Evaluation with Improved Human Correlation
Ananya Mukherjee and Manish Shrivastava

REUSE: REference-free UnSupervised Quality Estimation Metric
Ananya Mukherjee and Manish Shrivastava

MaTESe: Machine Translation Evaluation as a Sequence Tagging Problem
Stefano Perrella, Lorenzo Proietti, Alessandro Scirè, Niccolò Campolungo and Roberto Navigli

COMET-22: Unbabel-IST 2022 Submission for the Metrics Shared Task
Ricardo Rei, José G. C. De Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur and André F. T. Martins

Alibaba-Translate China's Submission for WMT2022 Metrics Shared Task
Yu Wan, Keqin Bao, Dayiheng Liu, Baosong Yang, Derek F. Wong, Lidia S. Chao, Wenqiang Lei and Jun Xie

16:00 - 17:30 *Quality Estimation Task*

Quality Estimation via Backtranslation at the WMT 2022 Quality Estimation Task
Sweta Agrawal, Nikita Mehandru, Niloufar Salehi and Marine Carpuat

Alibaba-Translate China's Submission for WMT 2022 Quality Estimation Shared Task
Keqin Bao, Yu Wan, Dayiheng Liu, Baosong Yang, Wenqiang Lei, Xiangnan He, Derek F. Wong and Jun Xie

KU X Upstage's Submission for the WMT22 Quality Estimation: Critical Error Detection Shared Task
Sugyeong Eo, Chanjun Park, Hyeonseok Moon, Jaehyung Seo and Heuiseok Lim

NJUNLP's Participation for the WMT2022 Quality Estimation Shared Task
Xiang Geng, Yu Zhang, Shujian Huang, Shimin Tao, Hao Yang and Jiajun Chen

BJTU-Toshiba's Submission to WMT22 Quality Estimation Shared Task
Hui Huang, Hui Di, Chunyou Li, Hanming Wu, Kazushige Ouchi, Yufeng Chen, Jian Liu and Jinan Xu

Papago's Submission to the WMT22 Quality Estimation Shared Task
Seunghyun Lim and Jeonghyeok Park

CometKiwi: IST-Unbabel 2022 Submission for the Quality Estimation Shared Task
Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. De Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie and André F. T. Martins

CrossQE: HW-TSC 2022 Submission for the Quality Estimation Shared Task
Shimin Tao, Su Chang, Ma Miaomiao, Hao Yang, Xiang Geng, Shujian Huang, Min Zhang, Jiaxin Guo, Minghan Wang and Yinglu Li

Welocalize-ARC/NKUA's Submission to the WMT 2022 Quality Estimation Shared Task
Eirini Zafeiridou and Sokratis Sofianopoulos

16:00 - 17:30 *Efficient Translaton Task*

Edinburgh's Submission to the WMT 2022 Efficiency Task
Nikolay Bogoychev, Maximiliana Behnke, Jelmer Van Der Linde, Graeme Nail, Kenneth Heafield, Biao Zhang and Sidharth Kashyap

CUNI Non-Autoregressive System for the WMT 22 Efficient Translation Shared Task
Jindřich Helcl

The RoyalFlush System for the WMT 2022 Efficiency Task
Bo Qin, Aixin Jia, Qiang Wang, Jianning Lu, Shuqin Pan, Haibo Wang and Ming Chen

HW-TSC's Submission for the WMT22 Efficiency Task
Hengchao Shang, Ting Hu, Daimeng Wei, Zongyao Li, Xianzhi Yu, Jianfei Feng, Ting Zhu, Lizhi Lei, Shimin Tao, Hao Yang, Ying Qin, Jinlong Yang, Zhiqiang Rao and Zhengzhe Yu

16:00 - 17:30 *Automatic Post-Editing Task*

IIT Bombay's WMT22 Automatic Post-Editing Shared Task Submission
Sourabh Deoghare and Pushpak Bhattacharyya

LUL's WMT22 Automatic Post-Editing Shared Task Submission
Xiaoying Huang, Xingrui Lou, Fan Zhang and Tu Mei

16:00 - 17:30 *Papers from the Findings of the EMNLP*

Translating Hanja Historical Documents to Contemporary Korean and English
Alice Oh, Kyunghyun Cho, JinYeong Bak, Haneul Yoo, Jiho Jin and Juhee Son

Data Selection Curriculum for Neural Machine Translation
Shafiq Joty, Shruti Bhosale, James Cross, Vishrav Chaudhary, Philipp Koehn and Tasnim Mohiuddin

Bitext Mining Using Distilled Sentence Representations for Low-Resource Languages
Holger Schwenk, Onur Çelebi and Kevin Heffernan

m^4 Adapter: Multilingual Multi-Domain Adaptation for Machine Translation with a Meta-Adapter
Alexander Fraser, Alexandra Chronopoulou and Wen Lai

What Do Compressed Multilingual Machine Translation Models Forget?
Laurent Besacier, James Henderson, Caroline Brun, Alexandre Berard, Vassilina Nikoulina and Alireza Mohammadshahi

Improving HowNet-Based Chinese Word Sense Disambiguation with Translations
Grzegorz Kondrak, Bradley Hauer and Xiang Zhang

Finding Memo: Extractive Memorization in Constrained Sequence Generation Tasks
Arul Menezes and Vikas Raunak

SALTED: A Framework for SAlient Long-tail Translation Error Detection
Arul Menezes, Matt Post and Vikas Raunak

SALTED: A Framework for SAlient Long-tail Translation Error Detection
Arul Menezes, Matt Post and Vikas Raunak

Wednesday, December 7, 2022 (continued)

Data Cartography for Low-Resource Neural Machine Translation
Marine Carpuat and Aquia Richburg

Improving Zero-Shot Multilingual Translation with Universal Representations and Cross-Mapping
Yang Feng and Shuhao Gu

Guiding Neural Machine Translation with Semantic Kernels
Yuqiang Xie, Luxi Xing, Yunpeng Li, Yubing Ren, Xiangpeng Wei, Yue Hu and Ping Guo

Does Simultaneous Speech Translation need Simultaneous Models?
Marco Turchi, Matteo Negri, Marco Gaido and Sara Papi

Utilizing Language-Image Pretraining for Efficient and Robust Bilingual Word Alignment
Kangwook Lee, Dimitris Papailiopoulos, Junjie Hu, Yifei Ming, Timothy Ossowski, Shashank Rajput, Jy-yong Sohn and Tuan Dinh

11:00 - 12:40 *Session 6 — Research Papers on Practical Aspects of Machine Translation*

Focused Concatenation for Context-Aware Neural Machine Translation
Lorenzo Lupo, Marco Dinarelli and Laurent Besacier

Does Sentence Segmentation Matter for Machine Translation?
Rachel Wicks and Matt Post

Revisiting Locality Sensitive Hashing for Vocabulary Selection in Fast Neural Machine Translation
Hieu Hoang, Marcin Junczys-dowmunt, Roman Grundkiewicz and Huda Khayrallah

Too Brittle to Touch: Comparing the Stability of Quantization and Distillation towards Developing Low-Resource MT Models
Harshita Diddee, Sandipan Dandapat, Monojit Choudhury, Tanuja Ganu and Kalika Bali

Data Augmentation for Inline Tag-Aware Neural Machine Translation
Yonghyun Ryu, Yoonjung Choi and Sangha Kim

Wednesday, December 7, 2022 (continued)

12:40 - 14:00	*Lunch Break*
14:00 - 15:30	*Session 7 — Invited Talk by Ondrej Bojar*

Thursday, December 8, 2022

09:00 - 09:10 *Session 5 — Shared Task Overview Papers II*

Findings of the WMT 2022 Biomedical Translation Shared Task: Monolingual Clinical Case Reports
Mariana Neves, Antonio Jimeno Yepes, Amy Siu, Roland Roller, Philippe Thomas, Maika Vicente Navarro, Lana Yeganova, Dina Wiemann, Giorgio Maria Di Nunzio, Federica Vezzani, Christel Gerardin, Rachel Bawden, Darryl Johan Estrada, Salvador Lima-lopez, Eulalia Farre-maduel, Martin Krallinger, Cristian Grozea and Aurelie Neveol

Findings of the WMT 2022 Shared Task on Chat Translation
Ana C Farinha, M. Amin Farajian, Marianna Buchicchio, Patrick Fernandes, José G. C. De Souza, Helena Moniz and André F. T. Martins

Findings of the First WMT Shared Task on Sign Language Translation (WMT-SLT22)
Mathias Müller, Sarah Ebling, Eleftherios Avramidis, Alessia Battisti, Michèle Berger, Richard Bowden, Annelies Braffort, Necati Cihan Camgöz, Cristina España-bonet, Roman Grundkiewicz, Zifan Jiang, Oscar Koller, Amit Moryossef, Regula Perrollaz, Sabine Reinhard, Annette Rios, Dimitar Shterionov, Sandra Sidler-miserez and Katja Tissi

Findings of the WMT'22 Shared Task on Large-Scale Machine Translation Evaluation for African Languages
David Adelani, Md Mahfuz Ibn Alam, Antonios Anastasopoulos, Akshita Bhagia, Marta R. Costa-jussà, Jesse Dodge, Fahim Faisal, Christian Federmann, Natalia Fedorova, Francisco Guzmán, Sergey Koshelev, Jean Maillard, Vukosi Marivate, Jonathan Mbuya, Alexandre Mourachko, Safiyyah Saleem, Holger Schwenk and Guillaume Wenzek

Findings of the WMT 2022 Shared Tasks in Unsupervised MT and Very Low Resource Supervised MT
Marion Weller-di Marco and Alexander Fraser

Overview and Results of MixMT Shared-Task at WMT 2022
Vivek Srivastava and Mayank Singh

Findings of the Word-Level AutoCompletion Shared Task in WMT 2022
Francisco Casacuberta, George Foster, Guoping Huang, Philipp Koehn, Geza Kovacs, Lemao Liu, Shuming Shi, Taro Watanabe and Chengqing Zong

Findings of the WMT 2022 Shared Task on Translation Suggestion
Zhen Yang, Fandong Meng, Yingxue Zhang, Ernan Li and Jie Zhou

10:30 - 11:00 *Coffee Break*

15:30 - 16:00 *Coffee Break*

16:00 - 17:30 *Session 8 — Shared Task Sytem Description Papers II*

16:00 - 17:30 *Biomedical Translation Task*

The SPECTRANS System Description for the WMT22 Biomedical Task
Nicolas Ballier, Jean-baptiste Yunès, Guillaume Wisniewski, Lichao Zhu and Maria Zimina

SRT's Neural Machine Translation System for WMT22 Biomedical Translation Task
Yoonjung Choi, Jiho Shin, Yonghyun Ryu and Sangha Kim

Examining Large Pre-Trained Language Models for Machine Translation: What You Don't Know about It
Lifeng Han, Gleb Erofeev, Irina Sorokina, Serge Gladkoff and Goran Nenadic

Summer: WeChat Neural Machine Translation Systems for the WMT22 Biomedical Translation Task
Ernan Li, Fandong Meng and Jie Zhou

Optum's Submission to WMT22 Biomedical Translation Tasks
Sahil Manchanda and Saurabh Bhagwat

Huawei BabelTar NMT at WMT22 Biomedical Translation Task: How We Further Improve Domain-specific NMT
Weixuan Wang, Xupeng Meng, Suqing Yan, Ye Tian and Wei Peng

HW-TSC Translation Systems for the WMT22 Biomedical Translation Task
Zhanglin Wu, Jinlong Yang, Zhiqiang Rao, Zhengzhe Yu, Daimeng Wei, Xiaoyu Chen, Zongyao Li, Hengchao Shang, Shaojun Li, Ming Zhu, Yuanchang Luo, Yuhao Xie, Miaomiao Ma, Ting Zhu, Lizhi Lei, Song Peng, Hao Yang and Ying Qin

16:00 - 17:30 *Chat Translation Task*

Unbabel-IST at the WMT Chat Translation Shared Task
João Alves, Pedro Henrique Martins, José G. C. De Souza, M. Amin Farajian and André F. T. Martins

Investigating Effectiveness of Multi-Encoder for Conversational Neural Machine Translation
Baban Gain, Ramakrishna Appicharla, Soumya Chennabasavaraj, Nikesh Garera, Asif Ekbal and Muthusamy Chelliah

BJTU-WeChat's Systems for the WMT22 Chat Translation Task
Yunlong Liang, Fandong Meng, Jinan Xu, Yufeng Chen and Jie Zhou

Thursday, December 8, 2022 (continued)

HW-TSC Translation Systems for the WMT22 Chat Translation Task
Jinlong Yang, Zongyao Li, Daimeng Wei, Hengchao Shang, Xiaoyu Chen, Zhengzhe Yu, Zhiqiang Rao, Shaojun Li, Zhanglin Wu, Yuhao Xie, Yuanchang Luo, Ting Zhu, Yanqing Zhao, Lizhi Lei, Hao Yang and Ying Qin

16:00 - 17:30 *Sign Language Translation Task*

Clean Text and Full-Body Transformer: Microsoft's Submission to the WMT22 Shared Task on Sign Language Translation
Subhadeep Dey, Abhilash Pal, Cyrine Chaabani and Oscar Koller

Spatio-temporal Sign Language Representation and Translation
Yasser Hamidullah, Josef Van Genabith and Cristina España-bonet

Experimental Machine Translation of the Swiss German Sign Language via 3D Augmentation of Body Keypoints
Lorenz Hufe and Eleftherios Avramidis

TTIC's WMT-SLT 22 Sign Language Translation System
Bowen Shi, Diane Brentari, Gregory Shakhnarovich and Karen Livescu

Tackling Low-Resourced Sign Language Translation: UPC at WMT-SLT 22
Laia Tarres, Gerard I. Gállego, Xavier Giro-i-nieto and Jordi Torres

16:00 - 17:30 *African Languages Translation Task*

Separating Grains from the Chaff: Using Data Filtering to Improve Multilingual Translation for Low-Resourced African Languages
Idris Abdulmumin, Michael Beukman, Jesujoba Alabi, Chris Chinenye Emezue, Everlyn Chimoto, Tosin Adewumi, Shamsuddeen Muhammad, Mofetoluwa Adeyemi, Oreen Yousuf, Sahib Singh and Tajuddeen Gwadabe

Language Adapters for Large-Scale MT: The GMU System for the WMT 2022 Large-Scale Machine Translation Evaluation for African Languages Shared Task
Md Mahfuz Ibn Alam and Antonios Anastasopoulos

Samsung Research Philippines - Datasaur AI's Submission for the WMT22 Large Scale Multilingual Translation Task
Jan Christian Blaise Cruz and Lintang Sutawika

University of Cape Town's WMT22 System: Multilingual Machine Translation for Southern African Languages
Khalid Elmadani, Francois Meyer and Jan Buys

Tencent's Multilingual Machine Translation System for WMT22 Large-Scale African Languages
Wenxiang Jiao, Zhaopeng Tu, Jiarui Li, Wenxuan Wang, Jen-tse Huang and Shuming Shi

DENTRA: Denoising and Translation Pre-training for Multilingual Machine Translation
Samta Kamboj, Sunil Kumar Sahu and Neha Sengupta

The VolcTrans System for WMT22 Multilingual Machine Translation Task
Xian Qian, Kai Hu, Jiaqiang Wang, Yifeng Liu, Xingyuan Pan, Jun Cao and Mingxuan Wang

WebCrawl African : A Multilingual Parallel Corpora for African Languages
Pavanpankaj Vegi, Sivabhavani J, Biswajit Paul, Abhinav Mishra, Prashant Banjare, Prasanna K R and Chitra Viswanathan

ANVITA-African: A Multilingual Neural Machine Translation System for African Languages
Pavanpankaj Vegi, Sivabhavani J, Biswajit Paul, Prasanna K R and Chitra Viswanathan

16:00 - 17:30 *Unsupervised and Very Low Resource Translation Task*

HW-TSC Systems for WMT22 Very Low Resource Supervised MT Task
Shaojun Li, Yuanchang Luo, Daimeng Wei, Zongyao Li, Hengchao Shang, Xiaoyu Chen, Zhanglin Wu, Jinlong Yang, Zhiqiang Rao, Zhengzhe Yu, Yuhao Xie, Lizhi Lei, Hao Yang and Ying Qin

Unsupervised and Very-Low Resource Supervised Translation on German and Sorbian Variant Languages
Rahul Tangsali, Aditya Vyawahare, Aditya Mandke, Onkar Litake and Dipali Kadam

MUNI-NLP Systems for Lower Sorbian-German and Lower Sorbian-Upper Sorbian Machine Translation @ WMT22
Edoardo Signoroni and Pavel Rychlý

The AIC System for the WMT 2022 Unsupervised MT and Very Low Resource Supervised MT Task
Ahmad Shapiro, Mahmoud Salama, Omar Abdelhakim, Mohamed Fayed, Ayman Khalafallah and Noha Adly

16:00 - 17:30 *Code-Mixed Translation Task*

NICT at MixMT 2022: Synthetic Code-Mixed Pre-training and Multi-way Fine-tuning for Hinglish–English Translation
Raj Dabre

Thursday, December 8, 2022 (continued)

The NiuTrans Machine Translation Systems for WMT22
Weiqiao Shan, Zhiquan Cao, Yuchen Han, Siming Wu, Yimin Hu, Jie Wang, Yi Zhang, Hou Baoyu, Hang Cao, Chenghao Gao, Xiaowen Liu, Tong Xiao, Anxiang Ma and Jingbo Zhu

MUCS@MixMT: IndicTrans-based Machine Translation for Hinglish Text
Asha Hegde and Shashirekha Lakshmaiah

SIT at MixMT 2022: Fluent Translation Built on Giant Pre-trained Models
Abdul Khan, Hrishikesh Kanade, Girish Budhrani, Preet Jhanglani and Jia Xu

The University of Edinburgh's Submission to the WMT22 Code-Mixing Shared Task (MixMT)
Faheem Kirefu, Vivek Iyer, Pinzhen Chen and Laurie Burchell

Findings of the WMT 2022 Biomedical Translation Shared Task: Monolingual Clinical Case Reports
Mariana Neves, Antonio Jimeno Yepes, Amy Siu, Roland Roller, Philippe Thomas, Maika Vicente Navarro, Lana Yeganova, Dina Wiemann, Giorgio Maria Di Nunzio, Federica Vezzani, Christel Gerardin, Rachel Bawden, Darryl Johan Estrada, Salvador Lima-lopez, Eulalia Farre-maduel, Martin Krallinger, Cristian Grozea and Aurelie Neveol

Domain Curricula for Code-Switched MT at MixMT 2022
Lekan Raheem, Maab Elrashid, Melvin Johnson and Julia Kreutzer

16:00 - 17:30 *Word-Level Autocompletion Task*

Lingua Custodia's Participation at the WMT 2022 Word-Level Auto-completion Shared Task
Melissa Ailem, Jingshu Liu, Jean-gabriel Barthelemy and Raheel Qader

Translation Word-Level Auto-Completion: What Can We Achieve Out of the Box?
Yasmin Moslem, Rejwanul Haque and Andy Way

PRHLT's Submission to WLAC 2022
Angel Navarro, Miguel Domingo and Francisco Casacuberta

IIGROUP Submissions for WMT22 Word-Level AutoCompletion Task
Cheng Yang, Siheng Li, Chufan Shi and Yujiu Yang

HW-TSC's Submissions to the WMT22 Word-Level Auto Completion Task
Hao Yang, Hengchao Shang, Zongyao Li, Daimeng Wei, Xianghui He, Xiaoyu Chen, Zhengzhe Yu, Jiaxin Guo, Jinlong Yang, Shaojun Li, Yuanchang Luo, Yuhao Xie, Lizhi Lei and Ying Qin

16:00 - 17:30 *Translation Suggestion Task*

TSMind: Alibaba and Soochow University's Submission to the WMT22 Translation Suggestion Task
Xin Ge, Ke Wang, Jiayi Wang, Nini Xiao, Xiangyu Duan, Yu Zhao and Yuqi Zhang

Transn's Submissions to the WMT22 Translation Suggestion Task
Mao Hongbao, Zhang Wenbo, Cai Jie and Cheng Jianwei

Improved Data Augmentation for Translation Suggestion
Hongxiao Zhang, Siyu Lai, Songming Zhang, Hui Huang, Yufeng Chen, Jinan Xu and Jian Liu

Findings of the 2022 Conference on Machine Translation (WMT22)

Tom Kocmi Microsoft
Rachel Bawden Inria, Paris
Ondřej Bojar Charles University
Anton Dvorkovich Neurodub
Christian Federmann Microsoft
Mark Fishel University of Tartu
Thamme Gowda Microsoft
Yvette Graham Trinity College Dublin
Roman Grundkiewicz Microsoft
Barry Haddow University of Edinburgh
Rebecca Knowles NRC
Philipp Koehn Johns Hopkins University
Christof Monz University of Amsterdam
Makoto Morishita NTT
Masaaki Nagata NTT
Toshiaki Nakazawa University of Tokyo
Michal Novák Charles University
Martin Popel Charles University
Maja Popović Dublin City University
Mariya Shmatova Neurodub

Abstract

This paper presents the results of the General Machine Translation Task organised as part of the Conference on Machine Translation (WMT) 2022. In the general MT task, participants were asked to build machine translation systems for any of 11 language pairs, to be evaluated on test sets consisting of four different domains. We evaluate system outputs with human annotators using two different techniques: reference-based direct assessment and (DA) and a combination of DA and scalar quality metric (DA+SQM).

1 Introduction

The Seventh Conference on Machine Translation (WMT22)[1] was held online with EMNLP 2022 and hosted a number of shared tasks on various aspects of machine translation. This conference built on 15 previous editions of WMT as workshops and conferences (Callison-Burch et al., 2007, 2008, 2009, 2010, 2011, 2012; Bojar et al., 2013, 2014, 2015, 2016, 2017, 2018; Barrault et al., 2019, 2020; Akhbardeh et al., 2021).

For more than a decade, the machine translation (MT) community has focused on the news domain, which has many desirable features for MT evaluation, such as sufficiently long and grammatically correct sentences that are easy for both professionals to translate (to produce references) and for human raters to evaluate without specific in-domain knowledge. However, with recent advances in MT and potential overfitting on the news domain (with methods such as fine-tuning on past WMT testsets), we decided to open a fresh research direction of testing the "General Machine Translation" capabilities.

How to test general MT capabilities is a research question in itself. Countless phenomena could be evaluated, the most important being:

- various domains (news, medicine, IT, patents, legal, social, gaming, etc.)
- style of text (formal or spoken language, fiction, technical reports, etc.)
- noisy or robust user-generated content (grammatical errors, code-switching, abbreviations, etc.)

Evaluating all possible phenomena is near impossible and creates many unforeseen problems. Therefore, we decided to simplify the problem and start with an evaluation of different domains. We select the following four domains: news, e-commerce, social, and conversational, chosen to represent various topics with different content

[1] http://www.statmt.org/wmt22/

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1 - 45
December 7-8, 2022 ©2022 Association for Computational Linguistics

styles. Additionally, these domains are understandable for humans without special in-domain knowledge, thus not requiring specialized translators or human raters for evaluation.

Another significant change for this year is the redesign of our human evaluation procedure for English→X and non-English language pairs. We introduce SQM-style DA rating, improved sampling of sentences for human judgements, and we opt in for using professional raters.

In addition to language pairs evaluated yearly, we introduce several new language pairs that have never been evaluated at WMT or other venues: Ukrainian↔English, Ukrainian↔Czech, Livonian↔English, Yakut↔Russian and English→Croatian.

Lastly, with multiple different shared tasks run at WMT evaluating different phenomena over the same language pairs, we proposed to aggregate test sets and ask participants of different shared tasks to also translate test sets from other shared tasks (for shared language pairs), allowing cross-task evaluation of systems on various phenomena. More details are in Section 4.2.

General MT task submissions and human judgements are available at `https://github.com/wmt-conference/wmt22-news-systems`. The interactive visualization and comparison of differences between systems is at `http://wmt.ufal.cz` using MT-ComparEval (Sudarikov et al., 2016).

The structure of the findings is as follows. We describe process of collecting, cleaning and translating of test sets in Section 2 followed by summary of allowed training data for constrained track Section 3. We list all submitted systems in Section 4. We use two different techniques for human evaluation. Reference-based DA is used to evaluate languages into English and described in Section 5. DA+SQM technique used for non-English and from English translation directions is described in Section 6. In Section 7, we describe our analysis of English→Croatian, translation direction containing professional and student produced references. We conclude the findings in Section 8.

2 Test Data

In this section, we describe the process of collecting data in Section 2.1, followed by the explanation of preprocessing steps in Section 2.2. Producing human references is summarized in Section 2.3 and test set analysis is conducted in Section 2.5. Lastly, Section 2.4 describes specific language pairs that are prepared differently.

2.1 Collecting test data

As in the news shared tasks in previous years, the test sets consist of unseen translations prepared specially for the task. However, in contrast, we introduce several domains instead of only the news domain. The test sets are publicly released to be used as translation benchmarks. Here we describe the production and composition of the test sets.

With the new direction towards testing general MT capabilities, we redesign the content of the test sets. We decided to collect data from four domains (news, social, e-commerce, and conversation). For all language pairs, we aimed for a test set size of 2000 sentences and to ensure that the test sets were "source-original", namely that the source text is written in the source language, and the target text is the human translation. This is to avoid "translationese" effects on the source language, which can have a detrimental impact on the accuracy of evaluation (Freitag et al., 2019; Läubli et al., 2020; Graham et al., 2020). We collected roughly the same number of sentences (around 500 sentences with document context) for each domain. For some languages, we could not locate high-quality data and therefore selected more sentences from other domains.

News domain - This domain contains data prepared in the same way as in previous years (Akhbardeh et al., 2021). We collect news articles from the second half of 2021 extracted from online news sites, keeping document information. The news domain is mainly of the highest quality.

Social domain - For most languages (Czech, English, French, German, and Japanese), we extract data from public Reddit discussions, keeping separate posts as a single document. We target subreddits that come from countries speaking a given language. We remove all posts marked by Reddit as inappropriate.

We use different data source for Chinese and Russian social domain as there is not enough Reddit content. For Chinese, we collected posts from various social media webpages used in China, a list provided by our Chinese colleague. For Russian, we took data from Zen, one of the most popular blog platforms among Russian-speaking users.

E-commerce domain - Contains product descriptions donated by individual companies.

			#segments			
Source / Domain	conversation	ecommerce	news	social	other	total
Chinese	349	518	505	503	-	1875
Czech	-	-	957	491	-	1448
Czech (to Ukrainian)	-	-	-	-	1930	1930
English	484	530	511	512	-	2037
English (to Croatian)	-	1015	656	-	-	1671
French	501	524	504	477	-	2006
German	462	501	506	515	-	1984
Japanese	502	503	505	498	-	2008
Livonian	-	-	-	-	420	420
Russian	-	508	1004	-	504	2016
Russian (to Yakut)	-	-	-	-	1123.0	1123
Ukrainian	-	-	-	-	2018	2018
Ukrainian (to Czech)	-	-	-	-	2812	2812
Yakut	-	-	-	-	1123	1123

Table 1: Number of segments for individual source languages used in the general translation test sets.

For Japanese e-commerce domain, we used search advertising text ads provided by an advertising company with their client's prior consent. Defining documents and sentences in search ads is tricky. Clients define multiple titles and multiple descriptions, called assets. We defined a document as the longest possible combination of assets. We also defined a sentence as either an asset or a unit separated by sentence-ending punctuations within an asset. Since the diversity of Japanese ad sentences is small, we chose the test sentences greedily to minimize the test set's self-BLEU.

Conversational domain - data for English, German, French, and Chinese are provided by the Chat Shared Task organizers (Farinha et al., 2022). These data contain a discussion between an agent, talking in English, and a customer, each of them talking in a different language. To avoid the effects of translationese, we split conversations into individual messages and handled each as a separate document, only using messages written in the original language (therefore, the English side only contains messages from agents) resulting in often short documents.

For Japanese conversational domain, We used question-answer pairs from a community question-answering website, *Oshiete!goo*[2]. The operator provided us with a dump as of March 2022. Topics are diverse, ranging from life advice to entertainment. Since there were usually many answers to a question, we extracted question-answer pairs whose answers were marked as the best answer. We considered a question-answer pair as a document and randomly sampled test data from question-answer pairs with a total length of 180 characters or fewer. We did not indicate the boundary between them.

After collecting all data, we applied several steps to filter out documents of lower-quality, see Section 2.2. Specifically paying attention to short documents. Whenever we had enough data, we removed the shortest documents, usually a single or two sentences. We advised linguists who were checking the data to further remove short documents. This helped us to add document context to the test set.

2.2 Human preprocessing of test data

In the News task of previous years, we asked humans to check collected data and carry out minor corrections (mainly checking sentence splits and discarding similar or repeated content), which was sufficient for the news domain because is often clean and without serious problems. However, with the expansion towards general MT, we run into an issue of source data being noisier and not well formatted that needs to be handled before translation.

Although testing of robustness of MT is an important task, the noisy data introduces problems for human translators and annotators. Therefore, we decided to discard data that are considered too noisy. Furthermore, publicly available data often contains inappropriate content, which can stress either human translators or human annotators, leading to a decrease in the quality (for example, translators refuse to translate political content considered censored in their countries).

Therefore, the source data for test sets[3] goes

[2] https://oshiete.goo.ne.jp

[3] Except for sources from the following translation directions: English→Croatian, Livonian↔English, Yakut↔Russian, Ukrainian→English, Ukrainian↔Czech. Data for these directions have been checked differently and should not contain noisy or inappropriate content.

through human validation checks involving linguists discarding inappropriate content altogether or carrying out minor textual corrections to the data. You can find the linguistic brief for prepossessing in Appendix C.

2.3 Test set translation

The translation of the test sets was performed by professional translation agencies, according to the brief in Appendix D. Different partners sponsored each language pair and various translation agencies were therefore used, which may affect the quality of the translation. The exception is that Chinese↔English, German↔English, Ukrainian↔English and reference-B for Czech↔English were translated by the same agency. These languages also received a special treatment of being translated by one translator and checked by a second different translator.

Several language pairs received special attention. For Chinese↔English, Czech↔English, German↔English, and English→Croatian, we obtained a second reference in each direction from different translators.

For Czech↔English, our partner paid professional agency to provide high-quality translations. However, as it turned out, the quality is rather low. We fixed manually the reference with grammar correction tools, however, that isn't sufficient. We provide this reference as reference–C. There is no issue with reference-B as that was provided by different partner.

Human translations would not be possible without the sponsorship of our partners. We are thankful for the support from: Microsoft, Charles University, LinguaCustodia, NTT, Dublin City University, Google, and Phrase.

2.4 Language pairs prepared differently

English→Croatian The English-Croatian test data is a sub-corpus of the DiHuTra corpus[4] (Lapshinova-Koltunski et al., 2022). The English source texts include Amazon product reviews and news articles. The document information is available for both domains.

The *reviews* were selected from the publicly available **Amazon product reviews**[5,6] containing reviews divided into 24 categories (topics). The selected corpus covers fourteen categories, paying attention to the data balance: an equal number of positive and negative reviews and a balanced distribution of categories (topics). In total, 196 reviews (1015 sentences) were included, fourteen from each of the fourteen selected topics: 'Beauty', 'Books', 'CDs and Vinyl', 'Cell Phones and Accessories', 'Grocery and Gourmet Food', 'Health and Personal Care', 'Home and Kitchen', 'Movies and TV', 'Musical Instruments', 'Patio, Lawn and Garden', 'Pet Supplies', 'Sports and Outdoors', 'Toys and Games' and 'Video Games'.

The *news articles* were selected from the News test corpus of the WMT (2019 and 2020) shared task.[7] In total, 68 news articles (656 sentences) from different sources are included.

These English texts were then translated into Croatian by professional translators and by translation students, thus providing two reference translations. Both professional and student translations were produced in cooperation with the University of Zagreb and the University of Rijeka in Croatia. In total, four professional translators and twenty translation students participated, all native speakers of Croatian and fluent in English. Translation experience of professional translators ranges between five and ten years, while for students the range is from zero to five years, the majority being in the range between two and four years. The two students who indicated no experience (zero years) also indicated that they had no real professional experience yet, only work in the framework of their studies. All students were in their first or second year of master's studies.

The translators were asked too keep the sentence (segment) alignment (not to merge or to split segments so that each English segment corresponds to one translated segment) and not to use any kind of machine translation in the process. No further restrictions were given to the translators.

Yakut↔Russian Source texts for Yakut↔Russian translation were selected from Ulus media, which is Yakutia's official news aggregator. The majority of the data are local news.[8] The professional translators were asked to translate 42 news texts for the test set. Yakut is one of the minor languages spoken by around

[4] https://github.com/katjakaterina/dihutra

[5] https://s3.amazonaws.com/amazon-reviews-pds/readme.html

[6] http://jmcauley.ucsd.edu/data/amazon/

[7] http://www.statmt.org/wmt20/translation-task.html

[8] https://ulus.media/

450,000 native speakers. It is one of the official languages of Sakha (Yakutia), a federal republic in the Russian Federation.

Livonian↔English The source language for Livonian↔English was English, since the amount of Livonian monolingual and parallel data is severely limited. The source texts were selected from various news articles published in 2022; politically neutral topics were selected. One addition to the set was the text describing the addition of WMT'22 Livonian↔English shared task itself. Translations were done by two professionals. Livonian is a critically endangered language spoken in Latvia but belonging to the Finno-ugric language family. Its last native speaker passed in 2013 and currently there are about 20 near-native speakers; however, there is an Institute of the Livonian Language at the University of Latvia that leads efforts on collecting and preserving Livonian texts as well as other materials (audio, video, hand-written, etc).

Ukrainian↔Czech and Ukrainian→English Source texts for Ukrainian↔Czech and Ukrainian→English translation were selected from the inputs collected through the Charles Translator for Ukraine.[9] Charles Translator for Ukraine is an online translation service that has been developed by the team from the Charles University, Prague[10] as a response to the wave of Ukrainian refugees coming to the Czech Republic after the 2022 Russian invasion of Ukraine.[11] The service is powered by a model trained with Block Backtranslation (Popel et al., 2020b). With users' consent, the service can log their inputs for the purpose of creating a dataset of real use cases. The datasets are extracted from the inputs collected in March and April 2022.

After automatic filtering,[12] we asked linguistically-educated annotators to filter and preprocess the source data manually. The filtering aimed at obtaining a data sample with diverse examples. The preprocessing was performed according to the brief in Appendix C with the following modifications. First, as the content is closely related to the war, someone may always find it polarizing or controversial. We did not filter out texts based on this criterion. Second, we asked the annotators not to delete or fix noisy inputs as long as they are comprehensible. This concerns, for instance, errors in casing, punctuation, diacritics, grammar and typos. Furthermore, all emojis are kept. Third, our annotators were instructed to join multiple related sentences to the same line whenever they found them too short compared to the rest of the dataset. The dataset thus does not satisfy the rule that each line contains a single sentence. Finally, any personal data related to people other than well-known people was pseudonymized.

The user inputs cover three broader domains: (1) personal communication, (2) news, and (3) formal communication. Our annotators assigned these categories (often accompanied by a finer subcategory) to every data example. If none of the above categories fit, they labeled the example with the "other" tag.

The source texts were translated by professional translation agencies principally following the brief in Appendix D. A sample of translated sentences were checked by native speakers of the target language. It revealed that post-edited MT had allegedly been used for parts of the Ukrainian→Czech test set, although this was denied by the translator. Therefore, we decided to add additional data to the test set for this direction translated by a different translation agency. This extra data consists of about 600 segments downloaded from the web (news, example CV) and about 200 segments from the Charles Translator inputs logs. It was pre-processed similarly as described above except for the domain annotation (all segments have the "unknown" tag assigned).

2.5 Test set analysis

As described previously, the aim was for the test sets to be composed of approximately 500 sentences per domain, although this depended on the language pair. The number of segments for each domain (including unspecified domain 'other') is given in Table 1 per source language, with the target language being specified where the composition differs. All four domains are available for Chinese, English, French, German and Japanese source texts, whereas only certain domains are available

[9] http://translate.cuni.cz

[10] http://ufal.mff.cuni.cz/u4u

[11] At that time, the most popular online MT services either did not support translation between Czech and Ukrainian (e.g. DeepL) or they seemed to pivot the translation for the language pair via English (e.g. Google Translate, Microsoft Translator).

[12] This includes the removal of intermediate inputs, HTML-tagged inputs, inputs identified as written in a language other than the source language, and backtranslated inputs.

for Czech, Russian and English into Croatian.

Document context Document context is available for most language pairs (the exception being Livonian↔English). The length of documents varies considerably by domain but also by language pair. As can be see in Table 2, e-commerce documents tend to be longest, followed by news and social (together), with conversational documents being shortest, although this does not hold for all languages. For example, the Ukrainian test set has short documents (2.28 segments on average), whereas Yakut↔Russian has very long ones (26.12 segments on average).

Lexical diversity We can compare the type-token ratio (TTR) to get an idea of the relative lexical diversity of (i) domains and (ii) original vs. translated sentences.[13],[14] Raw TTRs for each language pair and domain are given in Table 28 in Appendix E. Regarding domains, the TTR is generally lowest for conversations, whereas e-commerce and news are most diverse, followed by social. Translated texts appear to show a lower lexical diversity than original texts. If we look at the ratio between the TTRs of a language A and a language B (i.e. the diversity of A with respect to B), this ratio is higher when A is the source and B the target than when B is the source and A the target. For example, given the language pair Czech↔English, the ratio of the TTRs of Czech and English (i.e. $\frac{\text{TTR}_{\text{cs}}}{\text{TTR}_{\text{en}}}$ is higher when Czech is the original text and lower when it is the translation. This can be seen in Table 3 comparing for individual domains.

Anonymisation One characteristic that stands out is the presence of placeholders for anonymised elements in the conversation and social domains. There are a total of 17 difference placeholders, indicated by the entity type surrounded by #, e.g. #NAME#, #EMAIL#, #Product1#, #Product2#, etc. The entities are identical in the reference translation (where there is a direct translation), rather than the entity being translated (e.g. #NAME# and not #NOM# for French). Manual corrections were carried out to homogenise variants in terms of capitals, space issues and placeholders that were translated rather than copied by the professional translators.

Translation quality As mentioned previously, the quality of the human references differed according to the agency used. A few translations were erroneous due to problems with anonymisation, where some overzealous anonymisation added entity tags within non-entity words, therefore making the source sentence non-sensical. However this affected only one or two sentences, and some minor corrections were introduced. There were some particular problems with the quality of Czech→English translations, including wrong quote marks, grammatical and spelling mistakes and unnatural translations as mentioned in Section 2.3.

3 Training Data

Similar to previous years, we provide a selection of parallel and monolingual corpora for model training. The provenance and statistics of the selected parallel datasets are provided in Appendix in Table 20 and Table 19. Specifically, our parallel data selection include large multilingual corpora such as Europarl-v10 (Koehn, 2005), Paracrawl-v9 (Bañón et al., 2020), CommonCrawl, NewsCommentary-v16, WikiTitles-v3, WikiMatrix (Schwenk et al., 2021), TildeCorpus (Rozis and Skadiņš, 2017), OPUS (Tiedemann, 2012), UN Parallel Corpus (Ziemski et al., 2016), and language specific corpora such as CzEng-v2.0 (Kocmi et al., 2020), YandexCorpus,[15] ELRC EU Acts, YakutCorpus [16], JParaCrawl (Morishita et al., 2020), Japanese-English Subtitle Corpus (Pryzant et al., 2018), Livonian multiparallel corpus Liv4ever (Rikters et al., 2022),[17] KFTT(Neubig, 2011), TED (Cettolo et al., 2012), CCMT, and back-translated news. Similar to previous years, we provided links to these datasets on the task web page.[18] However, new to this year, we automate the data preparation pipeline using a tool named MTDATA (Gowda et al., 2021).[19] MTDATA downloads all available datasets, except

[13] The TTR is the ratio of unique tokens to total tokens, and it is higher the diverse the vocabulary of a text is. It is dependent on the morphological complexity of a language, but can also vary due to other factors.

[14] Texts are tokenised using the language-specific Spacy models (Honnibal and Montani, 2017) where available. For Czech, Livonian and Yakut, for which Spacy models are not available, we took as a rough approximation models for Croatian, Finnish and Russian respectively.

[15] https://github.com/mashashma/WMT2022-data

[16] https://github.com/mashashma/WMT2022-data/tree/main/yakut

[17] https://huggingface.co/datasets/tartuNLP/liv4ever

[18] https://statmt.org/wmt22/translation-task.html

[19] https://statmt.org/wmt22/mtdata

	#segments per doc					
Source / Domain	conversation	ecommerce	news	social	other	all
Chinese	2.13	17.86	13.29	20.12	-	7.32
Czech	-	-	14.07	7.12	-	10.57
Czech (to Ukrainian)	-	-	-	-	1.86	1.86
French	2.61	22.78	14.00	14.03	-	7.04
German	2.87	17.28	11.50	13.92	-	7.32
English	5.20	23.04	16.48	15.06	-	11.25
English (to Croatian)	-	5.18	9.65	-	-	6.33
Japanese	4.40	4.49	15.30	8.03	-	6.26
Livonian	-	-	-	-	1.00	1.00
Russian	-	10.58	12.55	-	5.09	8.88
Russian (to Yakut)	-	-	-	-	26.12	26.12
Ukrainian	-	-	-	-	2.28	2.28
Ukrainian (to Czech)	-	-	-	-	2.98	2.98
Yakut (to Russian)	-	-	-	-	26.12	26.12

Table 2: Average document length (in # segments) for individual source languages used in the general translation test sets.

	conversation		ecommerce		news		social		other	
Lang. pair	→	←	→	←	→	←	→	←	→	←
Czech–English	-	-	1.9	1.58	1.73	1.57	-			
Czech–Ukrainian	-	-	-	-	1.06	0.93				
German–English	1.39	1.00	1.50	1.13	1.35	1.15	1.38	1.13	-	
German–French	1.25	0.95	1.50	1.15	1.35	1.15	1.26	1.08	-	
English–Czech	-	-	0.63	0.52	0.64	0.58	-			
English–German	1.00	0.72	0.89	0.67	0.87	0.74	0.88	0.72	-	
English–Japanese	1.50	1.00	1.41	1.20	1.44	1.13	1.28	1.00	-	
English–Livonian	-	-	-	-	0.74	0.74				
English–Russian	-	0.69	0.59	0.67	0.57	-	-			
English–Chinese	1.15	0.71	1.09	0.70	1.00	0.68	0.92	0.74	-	
French–German	1.06	0.80	0.87	0.67	0.87	0.74	0.93	0.79	-	
Japanese–English	1.00	0.67	0.83	0.71	0.88	0.69	1.00	0.78	-	
Livonian–English	-	-	-	-	1.36	1.36				
Russian–English	-	1.69	1.46	1.75	1.50	-	-			
Russian–Yakut	-	-	-	-	0.89	0.89				
Yakut–Russian	-	-	-	-	1.12	1.12				
Ukrainian–Czech	-	-	-	-	1.08	0.94				
Chinese–English	1.41	0.87	1.43	0.92	1.47	1.00	1.35	1.09	-	

Table 3: For each language pair A–B, the ratio of the TTRs of A and B, for the A→B test set (→; i.e. A is the original text) and for the B→A test set (←, i.e. A is the translated text).

the two which required user authentication: CCMT and CzEng-v2.0.

4 System submissions

In 2022, we received a total of 107 primary submissions[20] and 82 online systems. The participating institutions are listed in Table 4 and detailed in the rest of this section. Each system did not necessarily appear in all translation tasks. We also included online MT systems (originating from 5 services), which we anonymized as ONLINE-A,B,G,W,Y. All submissions, sources and references are made available via github.

For presentation of the results, systems are treated as either constrained or unconstrained. When the system submitters report that they were only trained on our provided data, we class them as constrained. The online systems are treated as unconstrained during the automatic and human evaluations, since we do not know how they were built. In Appendix F, we provide brief details of the submitted systems, for those where the authors provided such details.

4.1 OCELoT

To collect submissions, we used the open-source OCELoT platform[21] again, which provides anonymized public leaderboards for several WMT22 shared tasks.[22] Similarly to the setup from the previous year, only registered and verified teams with correct contact information were allowed to submit their system outputs and each

[20] GTCOM was removed from human evaluation, however, we calculate automatic scores in Appendix G.

[21] https://github.com/AppraiseDev/OCELoT

[22] https://ocelot-wmt22.mteval.org

Team	Language Pairs	System Description
AISP-SJTU	en-ja, en-zh, ja-en, zh-en	Liu et al. (2022)
AIST	ja-en	(no associated paper)
ALMAnaCH-Inria	cs-en, cs-uk, ru-en, uk-cs, uk-en	Alabi et al. (2022)
AMU	cs-uk, uk-cs	Nowakowski et al. (2022)
ARC-NKUA	en-uk, uk-en	Roussis and Papavassiliou (2022)
CUNI-Bergamot	en-cs	Jon et al. (2022)
CUNI-DocTransformer	cs-en, en-cs	Jon et al. (2022)
CUNI-Transformer	cs-en, cs-uk, en-cs, uk-cs	Jon et al. (2022)
CharlesTranslator	cs-uk, uk-cs	Popel et al. (2022)
DLUT	en-ja, en-zh, ja-en, zh-en	(no associated paper)
GTCOM	cs-uk, en-hr, en-uk, en-zh, uk-cs, uk-en	Zong and Bei (2022)
HuaweiTSC	cs-uk, en-hr, en-liv, en-ru, en-uk, en-zh, liv-en, ru-en, uk-cs, uk-en, zh-en	Wei et al. (2022)
JDExploreAcademy	cs-en, de-en, en-cs, en-de, en-ja, en-ru, en-zh, ja-en, ru-en, zh-en	Zan et al. (2022)
KYB	en-ja, ja-en	Kalkar et al. (2022)
LT22	de-en, de-fr	Malli and Tambouratzis (2022)
Lan-Bridge	cs-en, cs-uk, de-en, en-cs, en-de, en-hr, en-ja, en-ru, en-uk, en-zh, fr-de, ja-en, ru-en, ru-sah, sah-ru, uk-cs, uk-en, zh-en	Han et al. (2022)
LanguageX	en-ja, en-zh, ja-en, zh-en	Zeng (2022)
Liv4ever	en-liv, liv-en	Rikters et al. (2022)
NAIST-NICT-TIT	en-ja, ja-en	Deguchi et al. (2022)
NT5	en-ja, ja-en	Morishita et al. (2022)
NiuTrans	en-hr, en-liv, liv-en, zh-en	Shan et al. (2022)
OpenNMT	en-de	(no associated paper)
PROMT	de-en, en-de, en-ru, uk-en	Molchanov et al. (2022)
SRPOL	en-hr, en-ru, ru-en	Dobrowolski et al. (2022)
TAL-SJTU	en-liv, liv-en	He et al. (2022)
TartuNLP	en-liv, liv-en	Tars et al. (2022)
eTranslation	en-ru, en-uk, fr-de	Oravecz et al. (2022)
manifold	en-zh	Jin et al. (2022)
shopline-pl	cs-en	(no associated paper)

Table 4: Participants in the shared translation task. The translations from the online systems were not submitted by their respective companies but were obtained by us, and are therefore anonymized in a fashion consistent with previous years of the workshop.

verified team was limited to 7 submissions per test set. Submissions on leaderboards with BLEU and CHRF scores from SacreBLEU (Post, 2018) were displayed anonymously to avoid publishing rankings based on automatic scores during the submission period. Until one week after the submission period, teams could select a single primary submission per test set, specify if the primary submission followed a constrained or unconstrained setting, and submit a system description paper abstract. All entries were mandatory for a system submission to be included in the human evaluation campaign.

OCELoT has helped to simplify the submission process—from collecting submissions to gathering system information—and it supported the multi-domain shift introduced in the general task this year. The platform was also used for the Biomedical Shared Task (Neves et al., 2022). This made it easier to include the biomedical test set as another domain data in the test sets of the general task for languages that overlapped between the two tasks, which made it possible to collect outputs from the general domain systems for the biomedical domain.

4.2 Collaboration across WMT shared tasks

There are various shared tasks at WMT evaluating same language pairs but with different participants. This leads into inability to compare systems specialized for a particular task with participants of other tasks.

Therefore, we decided to open a collaboration across WMT shared tasks by asking participants to translate test sets from other shared tasks as well. This open the possibility to see how general MT systems compete for example in biomedical domain, or what is the general translation quality of specialized systems.

We set up a collaboration with Biomedical Shared Task (Neves et al., 2022) on all shared language pairs (Chinese-English, German-English, Russian-English).

This effort did not increase the number of participants for General MT Task because all participants of Biomedical Shared Task also participated in General MT. However, other participants of General MT have been evaluated on biomedical domain, too. For details, see Neves et al. (2022).

Language Pair	Sys.	Assess.	Assess/Sys
Czech→English	12	20,094	1,674.5
German→English	10	21,006	2,100.6
Japanese→English	14	28,638	2,045.6
Livonian→English	5	4,638	927.6
Russian→English	10	27,651	2,765.1
Ukrainian→English	9	20,305	2,256.1
Chinese→English	13	28,120	2,163.1
Total to-English	73	150,452	2,061

Table 5: Amount of data collected in the WMT22 manual evaluation campaign for evaluation into-English; after removal of quality control items.

	All	(A) Sig. Diff. Bad Ref.	(A) & No Sig. Diff. Exact Rep.
Czech→English	373	91 (24%)	78 (86%)
German→English	365	92 (25%)	84 (91%)
Japanese→English	538	129 (24%)	113 (88%)
Livonian→English	101	15 (15%)	15 (100%)
Russian→English	601	140 (23%)	125 (89%)
Ukrainian→English	395	88 (22%)	83 (94%)
Chinese→English	395	98 (25%)	79 (81%)
Total	**1,422**	**428 (30%)**	**388 (91%)**

Table 6: Number of crowd-sourced workers taking part in the reference-based SR+DC campaign; (A) those whose scores for bad reference items were significantly lower than corresponding MT outputs; those of (A) whose scores also showed no significant difference for exact repeats of the same translation; note: many workers evaluated more than one language pair.

5 Human Evaluation of Translation into English

As in previous years, reference-based Direct Assessment (DA, Graham et al., 2013, 2014, 2016) was employed as the primary method of evaluation for translation into English. DA human evaluation has several important features including accurate quality control of crowd-sourcing and standard methods of significance testing differences in ratings for systems. Human assessors are asked to rate a given translation by how adequately it expresses the meaning of the corresponding reference translation or source language input on an analogue scale, which corresponds to an underlying absolute 0–100 rating scale.[23] Direct Assessment is also employed for evaluation of video captioning systems at TRECvid (Graham et al., 2018; Awad et al., 2019) and multilingual surface realisation (Mille et al., 2018, 2019, 2020).

For evaluation of translation into-English, we

[23] No sentence or document length restriction is applied during manual evaluation.

use the monolingual configuration of DA, where the human evaluator reads and rates the system output translation and compares its meaning to an English reference translation, which was manually translated by a human translator. As recommended in Graham et al. (2020), we only employ forward-created test data to avoid potential bias. Since evaluating segments without their context (i.e. the surrounding document) can cause further bias (Läubli et al., 2018; Toral et al., 2018), we evaluate sentences in turn taken from a single document and system (described as "SR+DC" in previous WMT reports).[24] Similarly to last year, for all language pairs for which document context was available, we include it when evaluating translations. Note that the ratings are nevertheless collected on the segment level, motivated by the power analysis described in Graham et al. (2020), as well as better inter-annotator agreement and lower effort described in Castilho (2020).

In terms of the manual evaluation for the translation task for into-English language pairs, a total of 428 Turker accounts were involved.[25] 510,451 translation assessment scores were submitted in total by the crowd, of which 187,922 were provided by workers who passed quality control.[26]

System rankings are produced from a large set of human assessments of translations, each of which indicates the absolute quality of the output of a system. Table 5 shows total numbers of human assessments collected in WMT22 for into-English language pairs contributing to final scores for systems.[27]

Quality control was carried out exactly as described in last year's WMT for crowd-sourcing into-English translation assessments on Amazon Mechanical Turk (see Akhbardeh et al. (2021) for full details). Table 6 shows results of workers who passed quality control (by showing significant differences in scores attributed to translations of known to be of distinct qualities) and numbers of workers who also showed no significant difference for ratings of identical pairs of translations judged separately in repeat tests. Data from the non-reliable workers in all language pairs were removed prior to calculation of results.

Similar to last year, all rankings for to-English translation were reached through segment ratings presented one at a time in their original document order (SR+DC). As is usual with DA assessments, human assessment scores for translations were first standardized according to each individual human assessor's overall mean and standard deviation score. Average standardized scores for individual segments belonging to a given system were then computed, before the final overall DA score for a given system is computed as the average of its segment scores (Ave z in Table 7). Results are also reported for average scores for systems, computed in the same way but without any score standardization applied (Ave % in Table 7).

Human performance estimates calculated through the evaluation of human-produced reference translations are denoted by "HUMAN" in all tables. Translations HUMAN-C in Czech→English are known to be of lower quality than usual for manual translations.

Clusters are identified by grouping systems together according to which systems significantly outperform all others in lower ranking clusters, according to Wilcoxon rank-sum test.

All data collected during the human evaluation is available at `http://www.statmt.org/wmt22/results.html`. Appendix B shows the official results for the underlying head-to-head significance test for all pairs of systems.

6 Human Evaluation of Translation out of English and without English

Human evaluation for out-of-English and non-English translation directions[28] was performed with source-based ("bilingual") direct assessment of individual segments in context similar to the approach described in Akhbardeh et al. (2021). We use open-source framework Appraise for the evaluation (Federmann, 2018).

This year, several changes were made to the annotation procedure, the data sampling, and the interface display. In contrast to the standard DA (sliding scale from 0-100) used in 2021, this year annotators performed DA+SQM (Direct Assessment + Scalar Quality Metric). In DA+SQM, the annotators still provide a raw score between 0 and 100, but also

[24] The implementation still has the limitation that the assessors cannot go back to the previous segment.

[25] Numbers do not include the 988 workers on Mechanical Turk who did not pass quality control.

[26] Both numbers include quality control segments.

[27] Number of systems for WMT22 includes "human" systems comprising human-generated reference translations used to provide human performance estimates.

[28] We decided not to run human evaluation for French↔German due to the small number of system submissions this year.

Czech→English

Rank	Ave.	Ave. z	System
1	74.0	0.133	Online-W
2	75.3	0.055	CUNI-DocTformer
2	69.8	0.050	Lan-Bridge
2	70.7	0.037	Online-B
2	72.5	−0.004	JDExploreAcad
2	70.5	−0.014	Online-A
2	71.2	−0.015	CUNI-Transformer
2	71.4	−0.028	Online-G
2	71.9	−0.086	SHOPLINE-PL
10	67.7	−0.145	Online-Y
11	61.2	−0.290	HUMAN-C
11	64.0	−0.301	ALMAnaCH-Inria

Japanese→English

Rank	Ave.	Ave. z	System
1	66.7	0.069	DLUT
1	66.1	0.068	NT5
1	66.3	0.059	JDExploreAcademy
1	67.0	0.054	LanguageX
1	68.2	0.049	Online-B
1	66.1	0.046	Online-W
1	68.5	0.016	Lan-Bridge
1	67.1	0.006	Online-G
1	64.8	0.006	Online-A
1	63.8	−0.018	AISP-SJTU
1	66.5	−0.021	NAIST-NICT-TIT
1	66.6	−0.035	Online-Y
1	62.5	−0.056	KYB
14	26.2	−1.285	AIST

Russian→English

Rank	Ave.	Ave. z	System
1	77.5	0.055	JDExploreAcademy
1	77.5	0.040	HuaweiTSC
1	75.0	0.033	Online-G
1	76.7	0.008	Lan-Bridge
1	75.2	0.005	Online-Y
1	74.6	−0.003	SRPOL
1	74.3	−0.011	Online-B
1	74.7	−0.021	Online-A
1	76.1	−0.039	Online-W
10	69.8	−0.238	ALMAnaCH-Inria

German→English

Rank	Ave.	Ave. z	System
1	68.8	0.004	Lan-Bridge
2	70.8	−0.023	Online-W
2	68.1	−0.038	JDExploreAcademy
2	64.1	−0.057	Online-G
2	67.3	−0.070	Online-A
2	68.3	−0.086	HUMAN-B
2	66.5	−0.089	Online-Y
2	66.3	−0.092	Online-B
2	64.8	−0.126	LT22
2	66.2	−0.127	PROMT

Ukrainian→English

Rank	Ave.	Ave. z	System
1	73.5	0.048	Lan-Bridge
1	74.8	0.047	Online-B
3	69.8	0.039	HuaweiTSC
3	69.8	0.007	Online-A
3	73.6	−0.010	PROMT
3	73.4	−0.023	Online-G
7	71.0	−0.071	Online-Y
7	70.2	−0.082	ARC-NKUA
9	68.8	−0.246	ALMAnaCH-Inria

Livonian→English

Rank	Ave.	Ave. z	System
1	67.7	0.024	TartuNLP
1	66.0	−0.014	TAL-SJTU
1	64.0	−0.035	HuaweiTSC
1	63.5	−0.079	Liv4ever
5	60.4	−0.346	NiuTrans

Chinese→English

Rank	Ave.	Ave. z	System
–	73.4	0.134	HUMAN-B
1	69.8	−0.026	JDExploreAcademy
1	69.0	−0.034	HuaweiTSC
1	69.1	−0.063	AISP-SJTU
1	69.2	−0.079	LanguageX
1	69.7	−0.083	Online-A
1	68.6	−0.083	DLUT
1	67.4	−0.089	Online-B
1	69.9	−0.098	Online-G
1	66.5	−0.109	Online-W
1	65.3	−0.117	Lan-Bridge
1	66.5	−0.122	Online-Y
1	66.3	−0.164	NiuTrans

Table 7: Official results of WMT22 General Translation Task for translation into-English (SR+DC). Systems ordered by DA score z-score; systems within a cluster are considered tied; lines indicate clusters according to Wilcoxon rank-sum test $p < 0.05$; rank ranges are based on the same test; grayed entry indicates resources that fall outside the constraints provided.

are presented with seven labeled tick marks, as visible in Figure 1. Discrete SQM (0-6) was found to correlate well with MQM (Multidimensional Quality Metrics) annotations by Freitag et al. (2021), while internal preliminary experiments suggested that DA+SQM helps to stabilize scores across annotators (as compared to DA). Annotators performing DA+SQM annotations at IWSLT 2022 human evaluation campaign (Anastasopoulos et al., 2022) also provided positive feedback about the annotation format. In previous years, full documents were sampled for annotation. This year we sampled a maximum of 10 consecutive segments from a document (a document "snippet") for annotation. This provides the potential to annotate segments from a more diverse range of documents while still maintaining a similar number of total annotations. Up to 10 source segments preceding and following the snippet being evaluated are displayed as static extra context for the annotator in the interface, as presented in Figure 1. As in past years, annotators provide both segment-level scores and document-level scores (in this case it is more accurate to call them snippet-level scores), however only the segment-level scores were used to compute the official rankings. As the English–Livonian data was not document-level, those annotations are run with segment-level-only DA+SQM. HITs (using the Amazon terminology of "Human Intelligence Task" to describe an annotation task) contained quality control segments, as described in Section 6.2. Rankings are computed as described in Section 6.4 based on segment-level scores.

6.1 Human Annotators

All annotations in the bilingual human evaluation campaign were carried out by hired professional annotators. This year, for the first time, we did not ask participants of the general task to contribute to human evaluation, but instead made it voluntary. The main motivations for this change were the attempt to increase the reliability and consistency of the judgements and the immense amount of time that was needed to be devoted to the process of collecting annotations from participating teams. Annotations for different language pairs were provided by different parties with their pool of annotators of distinct profiles as summarized in Table 8.

Charles University provided annotators for language pairs involving the Czech language, i.e. English→Czech and Ukrainian↔Czech. Their annotators were linguists, translators, researchers and students who are native speakers of the target language[29] with high proficiency in the source language.

University of Tartu provided the annotations for Livonian↔English, with 15% of the Livonian-speaking population participating in the annotation efforts. All three participants were near-native speakers of Livonian and participated in source-based Livonian-English and English-Livonian annotations, as well as reference-based Livonian annotation.

The second annotator group was provided by Toloka AI,[30] who collected annotations for English→Russian and Russian↔Yakut. Toloka AI is a global data labeling company that helps its customers to generate machine learning data at scale by harnessing the wisdom of the crowd from around the world. It relies on a geographically diverse crowd of several million registered users[31] (Pavlichenko et al., 2021). Toloka tests proficiency of their annotator crowd and excludes from future annotations anyone who does not pass quality control in the Appraise tool.

The last part of annotations was sponsored by Microsoft, who contributed with their pool of qualified paid bilingual speakers experienced in the MT evaluation process. Microsoft provided annotations for English into Chinese, Croatian, German, and Japanese, as well as Chinese→English as a comparison for reference-based evaluation described above and MQM evaluated in Metrics shared task (Freitag et al., 2022). For this pool of annotators, their performance is tracked over time, and those who fail quality control are permanently removed from the pool. This process increases the overall quality of the human assessment.

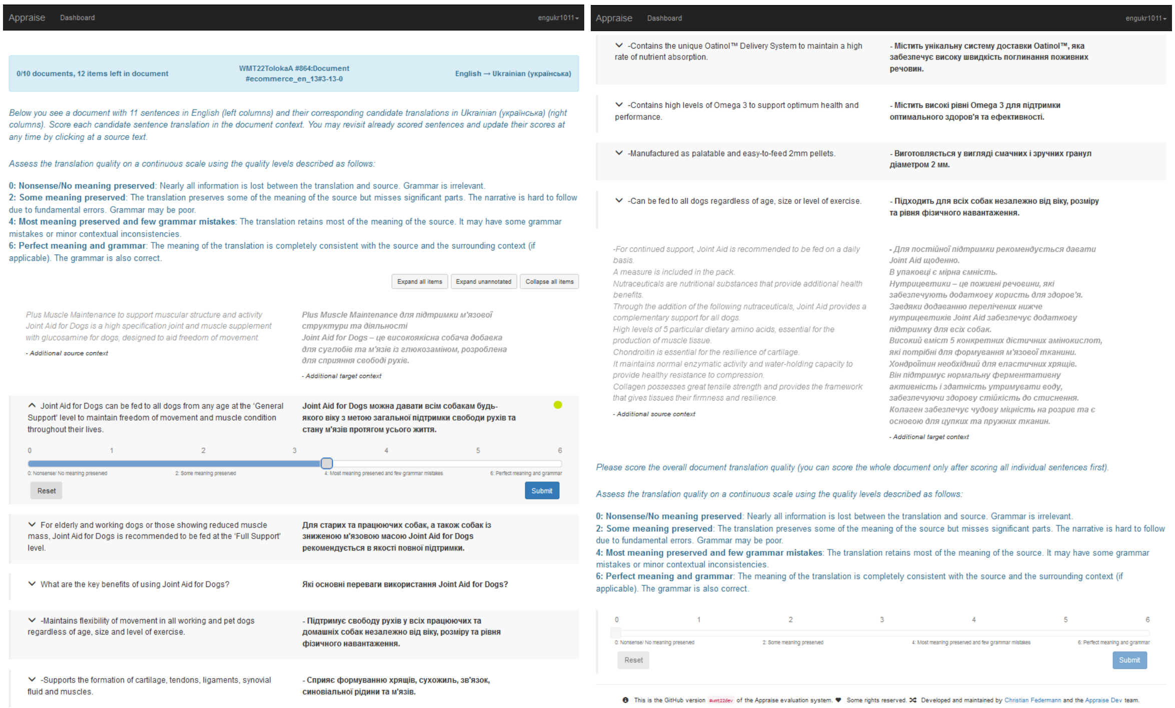
6.2 Sampling and Quality Control

In past WMT annotations, document-system pairs were sampled randomly for annotation, resulting in different subsets of the test set being annotated for each system. This year we first randomly sample a subset of document snippets from each of the domains for annotations, sampling the domains

[29] Some of Ukrainian→Czech annotators were not native Czechs, but native Ukrainians with near-native knowledge of Czech.

[30] https://toloka.ai

[31] https://hackernoon.com/evolution-of-the-data-production-paradigm-in-ai


Appraise
Dashboard
engukr1011
Expand all items
Expand unannotated
Collapse all items
0
1
2
3
4
5
6
0: Nonsense/ No meaning preserved
2: Some meaning preserved
4: Most meaning preserved and few grammar mistakes
6: Perfect meaning and grammar
Reset
Submit

0/10 documents, 12 items left in document

WMT22TolokaA #864:Document #ecommerce_en_13#3-13-0

English → Ukrainian (українська)

Below you see a document with 11 sentences in English (left columns) and their corresponding candidate translations in Ukrainian (українська) (right columns). Score each candidate sentence translation in the document context. You may revisit already scored sentences and update their scores at any time by clicking at a source text.

Assess the translation quality on a continuous scale using the quality levels described as follows:

0: Nonsense/No meaning preserved: Nearly all information is lost between the translation and source. Grammar is irrelevant.
2: Some meaning preserved: The translation preserves some of the meaning of the source but misses significant parts. The narrative is hard to follow due to fundamental errors. Grammar may be poor.
4: Most meaning preserved and few grammar mistakes: The translation retains most of the meaning of the source. It may have some grammar mistakes or minor contextual inconsistencies.
6: Perfect meaning and grammar: The meaning of the translation is completely consistent with the source and the surrounding context (if applicable). The grammar is also correct.

Plus Muscle Maintenance to support muscular structure and activity Joint Aid for Dogs is a high specification joint and muscle supplement with glucosamine for dogs, designed to aid freedom of movement. *- Additional source context*	*Plus Muscle Maintenance для підтримки м'язової структури та діяльності Joint Aid for Dogs – це високоякісна собача добавка для суглобів та м'язів із глюкозаміном, розроблена для сприяння свободі рухів.* *- Additional target context*
Joint Aid for Dogs can be fed to all dogs from any age at the 'General Support' level to maintain freedom of movement and muscle condition throughout their lives.	**Joint Aid for Dogs можна давати всім собакам будь-якого віку з метою загальної підтримки свободи рухів та стану м'язів протягом усього життя.**
For elderly and working dogs or those showing reduced muscle mass, Joint Aid for Dogs is recommended to be fed at the 'Full Support' level.	**Для старих та працюючих собак, а також собак із зниженою м'язовою масою Joint Aid for Dogs рекомендується в якості повної підтримки.**
What are the key benefits of using Joint Aid for Dogs?	**Які основні переваги використання Joint Aid for Dogs?**
-Maintains flexibility of movement in all working and pet dogs regardless of age, size and level of exercise.	**- Підтримує свободу рухів у всіх працюючих та домашніх собак незалежно від віку, розміру та рівня фізичного навантаження.**
-Supports the formation of cartilage, tendons, ligaments, synovial fluid and muscles.	**- Сприяє формуванню хрящів, сухожиль, зв'язок, синовіальної рідини та м'язів.**
-Contains the unique Oatinol™ Delivery System to maintain a high rate of nutrient absorption.	**- Містить унікальну систему доставки Oatinol™, яка забезпечує високу швидкість поглинання поживних речовин.**
-Contains high levels of Omega 3 to support optimum health and performance.	**- Містить високі рівні Omega 3 для підтримки оптимального здоров'я та ефективності.**
-Manufactured as palatable and easy-to-feed 2mm pellets.	**- Виготовляється у вигляді смачних і зручних гранул діаметром 2 мм.**
-Can be fed to all dogs regardless of age, size or level of exercise.	**- Підходить для всіх собак незалежно від віку, розміру та рівня фізичного навантаження.**
-For continued support, Joint Aid is recommended to be fed on a daily basis. A measure is included in the pack. Nutraceuticals are nutritional substances that provide additional health benefits. Through the addition of the following nutraceuticals, Joint Aid provides a complementary support for all dogs. High levels of 5 particular dietary amino acids, essential for the production of muscle tissue. Chondroitin is essential for the resilience of cartilage. It maintains normal enzymatic activity and water-holding capacity to provide healthy resistance to compression. Collagen possesses great tensile strength and provides the framework that gives tissues their firmness and resilience. *- Additional source context*	*- Для постійної підтримки рекомендується давати Joint Aid щоденно. В упаковці є мірна ємність. Нутрицевтики – це поживні речовини, які забезпечують додаткову користь для здоров'я. Завдяки додаванню перелічених нижче нутрицевтиків Joint Aid забезпечує додаткову підтримку для всіх собак. Високий вміст 5 конкретних дієтичних амінокислот, які потрібні для формування м'язової тканини. Хондроїтин необхідний для еластичних хрящів. Він підтримує нормальну ферментативну активність і здатність утримувати воду, забезпечуючи здорову стійкість до стиснення. Колаген забезпечує чудову міцність на розрив та є основою для цупких та пружних тканин.* *- Additional target context*

Please score the overall document translation quality (you can score the whole document only after scoring all individual sentences first).

Assess the translation quality on a continuous scale using the quality levels described as follows:

0: Nonsense/No meaning preserved: Nearly all information is lost between the translation and source. Grammar is irrelevant.
2: Some meaning preserved: The translation preserves some of the meaning of the source but misses significant parts. The narrative is hard to follow due to fundamental errors. Grammar may be poor.
4: Most meaning preserved and few grammar mistakes: The translation retains most of the meaning of the source. It may have some grammar mistakes or minor contextual inconsistencies.
6: Perfect meaning and grammar: The meaning of the translation is completely consistent with the source and the surrounding context (if applicable). The grammar is also correct.

This is the GitHub version #wmt22dev of the Appraise evaluation system. Some rights reserved. Developed and maintained by Christian Federmann and the Appraise Dev team.

(a) Top part of the screen with segment-level scoring. **(b)** Bottom part of the screen with document-level scoring.

Figure 1: Screen shot of the document-level DA+SQM configuration in the Appraise interface for an example assessment from the human evaluation campaign for out of English language pairs. The annotator is presented with the entire translated document snippet randomly selected from competing systems (anonymized) with preceding and following contexts and is asked to rate the translation of individual segments and then the entire document on sliding scales.

Language pairs	Annotators' profile
English→Chinese/Croatian/German/Japanese	Microsoft annotators: bilingual target-language native speakers, professional translators or linguists, experienced in MT evaluation
English→Czech	Czech paid linguists, annotators, researchers, students with high proficiency in English
English→Livonian	Livonian speakers
English→Russian/Ukrainian, Russian↔Yakut	Toloka paid crowd: bilingual target-language native speakers
Ukrainian↔Czech	Paid translators and target-language native speakers

Table 8: Human annotator types for each language pair in bilingual human evaluation.

with approximately the same number of segments per domain. We use document snippets with 10 consecutive segments, or fewer in the case of short documents. In this way, all systems are annotated over almost exactly the same subset of document snippets.[32] All HITs consists of exactly 100 segments and are generated as in the past: (1) first snippet-system pairs are randomly sampled (from the restricted set of pre-sampled snippets) with up to 80 segments; (2) then random snippets with the remaining 20 (or more) segments are duplicated to serve as quality control items; (3) BAD references are introduced to the random segments in the duplicated snippets to have about 12-14% of quality control segments per HIT.[33] BAD references consist of segments in which an embedded sequences of tokens is replaced from a randomly placed phrase of the same length, sampled from a different reference segment.

We perform quality control by measuring an annotator's ability to reliably score BAD translations significantly lower than corresponding original system outputs using a paired significance test with $p < 0.05$. We pair two HITs into a single annota-

[32] For English→{Czech, German, Japanese, Russian, Ukrainian, Chinese} and the additional Chinese→English collection, all systems received annotations for all the sampled snippets. For Czech→Ukrainian, Ukrainian→Czech, English→Croatian, Yakut→Russian, and pairs including Livonian, annotation coverage of sampled snippets was incomplete; not all systems were scored over exactly the same set of segments.

[33] For full details, see the HIT and batch generation code: https://github.com/wmt-conference/wmt22-news-systems

Language Pair	Sys.	Assess.	Assess/Sys
Chinese→English	14	26,800	1,914.3
Czech→Ukrainian	12	21,285	1,773.8
English→Czech	12	24,000	2,000.0
English→German	11	21,800	1,981.8
English→Croatian	10	19,046	1,904.6
English→Japanese	14	27,600	1,971.4
English→Livonian	6	3,903	650.5
English→Russian	12	46,675	3,889.6
English→Ukrainian	9	35,048	3,894.2
English→Chinese	14	27,800	1,985.7
Yakut→Russian	3	4,200	1,400.0
Ukrainian→Czech	12	14,622	1,218.5

Table 9: Amount of data collected in the WMT22 manual evaluation campaign for evaluation out-of-English; including human references as systems; after removal of quality control items.

Language Pair	Ann.	HITs	HITs/Ann.
Chinese→English	12	134	11.2
English→Czech	16	120	7.5
English→German	14	109	7.8
English→Croatian	13	96	7.4
English→Japanese	17	138	8.1
English→Chinese	8	139	17.4

Table 10: Numbers of individual annotators taking part in the WMT22 human evaluation campaign and the average number of HITs collected per annotator.

tion task with about 24-28 quality control segments to ensure a sufficient sample size for the statistical test. If an annotator is not able to demonstrate reliability on BAD references, they are excluded from further annotations, the HITs are reset and annotated from scratch by another annotator.[34]

In addition to the quality control items, because this annotation is performed bilingually, reference translations are also evaluated as though they were submitted systems.

For language pairs where there was a concern about having sufficient annotations, two smaller batches of HITs were generated (such that at least all segments in the first batch could be covered for all systems, with the second campaign completed if possible; in the case of translation between Czech and Ukrainian, due to a large number of single-sentence documents, larger documents were sampled first).

Language Pair	Min.	Max.	Med.
Chinese→English	0.03	0.77	0.40
English→Czech	0.15	0.81	0.49
English→German	-0.18	0.47	0.21
English→Croatian	0.23	0.65	0.41
English→Japanese	-0.11	0.68	0.24
English→Chinese	-0.13	0.56	0.16

Table 11: Minimum, maximum, and median Spearman's rank correlation coefficients between pairs of annotators on calibration HIT segments.

Source-Based English→Livonian (Official WMT22 ranking)

Rank	Ave.	Ave. z	System
1	74.4	1.255	HUMAN-A
2	46.2	0.215	TAL-SJTU
3-4	36.9	-0.147	HuaweiTSC
3-4	36.3	-0.175	TartuNLP
5	33.8	-0.262	Liv4ever
6	17.9	-0.853	NiuTrans

Ref.-Based English→Livonian

Rank	Ave.	Ave. z	System
1	39.5	0.499	TAL-SJTU
2-4	31.8	0.077	TartuNLP
2-4	31.5	0.051	Liv4ever
2-4	31.0	0.037	HuaweiTSC
5	18.3	-0.656	NiuTrans

Source-Based Livonian→English

Rank	Ave.	Ave. z	System
1	81.7	1.009	HUMAN-A
2-3	60.3	0.257	TartuNLP
2-3	60.2	0.252	TAL-SJTU
4	50.4	-0.084	HuaweiTSC
5	41.3	-0.406	Liv4ever
6	23.1	-1.052	NiuTrans

Table 12: Three rankings for systems translating between English and Livonian.

6.3 Calibration HITs

For several language pairs (English→{Chinese, Croatian, Czech, German, Japanese} and Chinese→English), we collect calibration HITs in the DA+SQM interface: one identical HIT with 100 randomly selected segments completed by all annotators, in addition to their regular annotation HITs. By providing a small set of sentences annotated by all annotators, we are better able to examine questions about inter-annotator consistency. We release these alongside the other annotations and the anonymized mapping between annotators and HITs in order to enable additional analysis.

Table 10 shows the number of unique annotators for these languages, along with the total number of HITs and average number of HITs per annotator. For all pairs of annotators who completed both a calibration HIT and additional HIT(s) within a given language pair, we compute the Spearman's rank correlation coefficient between the two an-

[34] The quality control in bilingual human evaluation excluded 17 HITs in total: 1 Yakut→Russian, 2 English→Russian, 3 English→Ukrainian, 7 English→Livonian, 4 Czech↔Ukrainian.

notators' scores of the segments in the calibration HIT. Table 11 shows the minimum, maximum, and median correlations obtained by pairs of annotators for each language. These vary quite widely between languages, and we also note that across the calibration HITs, annotators vary widely in their use of the scoring space and the shape of their score distributions. Even within the same language pair (i.e., scoring the exact same set of segments in the calibration HIT), some annotators' scores are distributed across most of the 0-100 scoring space, some only produce scores above a certain threshold, and some treat the scale as though it were discretized according to the numerical scale shown in the interface (clustering most of their scores at the numerical marks the one can see in Figure 1).

6.4 Human Ranking Computation

The official rankings shown in Table 13 are generated on the basis of the segment-level DA+SQM scores that are collected within document snippet context for all language pairs.[35] The quality control (BAD) segments and any HITs that failed to pass quality control are removed prior to computing the rankings. Means and standard deviations for computing z-scores are computed at the HIT level. To compute system-level averages (both raw and z-score), any instances of multiple scores for the same segment are first averaged together, then all segment-level scores are averaged per system to compute the system-level scores. The clusters are computed using the Wilcoxon rank-sum test with $p < 0.05$. Rank ranges indicate the number of systems a particular system underperforms or outperforms (i.e., the top end of the rank range is $l + 1$ where l is the number of losses, while the bottom is $n - w$ where n is the total number of systems and w is the number of systems that the system in questions significantly wins against).

The rankings for translation between Livonian and English shown in Table 12 are computed in the same manner described above, but because the test set does not include document boundaries the data was collected without document context and some of the data collection was source-based while other portions were reference-based. As the official ranking for English→Livonian we consider the ranking computed from source-based human evaluation.

6.5 Comparison of Human Evaluation Methods

In collaboration with the metrics shared task (Freitag et al., 2022), human annotation data for the Chinese→English direction was collected using three different approaches: the official monolingual reference-based SR+DC DA (Section 5, Table 7), the source-based fully document-level DA+SQM approach used for out-of-English and non-English directions (Section 6), and the Multidimensional Quality Metrics (MQM) framework (Freitag et al., 2021, 2022). We present the rankings produced by the three approaches in Table 14.

The DA rankings produced large clusters only for this language pair; that is, it was not possible to separate the performance into many system clusters with statistical significance. It is also important to note that the set of data over which each of these rankings was produced may have differed (e.g., the distribution over topic domains or the amount of coverage of the full test set), making it difficult to determine whether these differences in rankings represent differences due to data or due to different annotation methods.

7 Manual Error Analysis of English→Croatian translations

In addition to the official human evaluation by assigning DA scores, an analysis of errors in English→Croatian translations was carried out by an MT researcher with experience in human translation. The evaluation was carried out bilingually, while looking at the original English segment and all of its translations, both machine and human, all mixed together in a random order. The segments were presented in the natural order in the document, and the entire document (news article or review) was available by scrolling down or up.

The analysis was performed on the first 100 documents (80 reviews and 20 news articles), containing 603 segments (416 in reviews and 187 in news). All 14 review topics mentioned in Section 2.4 are included, although not uniformly distributed. The annotations are publicly available at https://github.com/wmt-conference/wmt22-news-systems/humaneval/en-hr/.

The errors were not coupled to any quality criterion (adequacy, fluency, readability) – all problematic words found in the translations were tagged as errors, no matter whether they are related to the source language, or are specific to the target lan-

[35] The code used to generate the rankings in Table 13 can be found here: https://github.com/AppraiseDev/Appraise/blob/main/Campaign/management/commands/ComputeWMT21Results.py

English→Czech

Range	Ave.	Ave. z	System
1	91.2	0.335	HUMAN–C
2	90.9	0.279	Online-W
3	88.6	0.158	JDExploreAcad.
4-6	85.3	0.045	Online-B
4-6	87.1	0.041	Lan-Bridge
4-6	85.1	0.029	HUMAN-B
7-10	84.2	−0.059	CUNI-Bergamot
7-10	83.7	−0.074	CUNI-DocTransf.
7-10	84.0	−0.087	Online-A
7-10	83.2	−0.128	CUNI-Transf.
11-12	83.3	−0.258	Online-G
11-12	80.8	−0.310	Online-Y

Czech→Ukrainian

Range	Ave.	Ave. z	System
1	85.6	0.295	HUMAN-A
2-5	84.6	0.225	Online-B
2-3	84.1	0.151	AMU
3-6	82.5	0.125	Lan-Bridge
3-6	81.1	0.065	HuaweiTSC
4-8	81.9	0.062	CharlesTranslator
6-8	80.2	0.026	CUNI-JL-JH
6-8	80.2	−0.002	CUNI-Transf.
9-10	79.8	−0.008	Online-G
9-10	79.2	−0.075	Online-A
11	76.0	−0.257	Online-Y
12	68.4	−0.669	ALMAnaCH-Inria

Ukrainian→Czech

Range	Ave.	Ave. z	System
1	89.6	0.417	HUMAN-A
2-3	85.6	0.182	AMU
2-4	83.5	0.148	HuaweiTSC
4-8	83.5	0.127	Lan-Bridge
3-8	82.0	0.110	CUNI-Transf.
4-8	82.5	0.082	CharlesTranslator
4-8	81.4	0.052	CUNI-JL-JH
4-8	81.9	0.042	Online-B
9-10	80.0	-0.101	Online-A
9-10	77.5	-0.138	Online-G
11	73.9	-0.351	Online-Y
12	69.2	-0.617	ALMAnaCH-Inria

Yakut→Russian

Range	Ave.	Ave. z	System
1	71.3	0.708	HUMAN-A
2	54.6	0.178	Online-G
3	16.0	−0.873	Lan-Bridge

English→Chinese

Range	Ave.	Ave. z	System
1	81.7	0.154	HUMAN-A
2-5	81.9	0.099	Online-W
2-5	80.9	0.074	HUMAN-B
2-9	80.3	0.073	JDExploreAcad.
2-7	79.7	0.026	Online-Y
4-11	80.0	0.020	Lan-Bridge
4-11	78.5	0.019	Manifold
5-12	79.4	−0.012	LanguageX
5-12	79.4	−0.019	Online-B
6-12	78.7	−0.020	Online-A
8-12	79.6	−0.043	HuaweiTSC
6-12	79.0	−0.045	AISP-SJTU
13-14	77.5	−0.150	DLUT
13-14	77.2	−0.153	Online-G

English→German

Range	Ave.	Ave. z	System
1-6	93.9	0.116	HUMAN-A
1-4	93.6	0.106	Online-B
1-4	93.4	0.106	Online-W
1-5	92.4	0.071	JDExploreAcad.
3-7	93.8	0.051	HUMAN-B
5-9	93.6	0.015	Lan-Bridge
4-9	91.1	−0.019	Online-A
6-11	92.2	−0.054	Online-Y
6-11	93.2	−0.066	Online-G
8-11	90.8	−0.110	PROMT
8-11	89.9	−0.189	OpenNMT

English→Japanese

Range	Ave.	Ave. z	System
1	86.3	0.218	HUMAN-A
2-11	84.1	0.103	NT5
2-9	83.6	0.099	LanguageX
2-9	84.3	0.093	JDExploreAcad.
2-8	84.3	0.087	Online-B
2-9	83.9	0.078	DLUT
2-11	83.2	0.058	Online-Y
3-11	82.9	0.022	Lan-Bridge
6-11	82.9	0.018	Online-A
2-11	83.3	0.004	NAIST-NICT-TIT
11-12	81.9	−0.027	AISP-SJTU
6-12	83.0	−0.029	Online-W
13	79.5	−0.311	Online-G
14	76.9	−0.434	KYB

English→Russian

Range	Ave.	Ave. z	System
1-2	87.3	0.222	Online-W
1-2	86.6	0.194	HUMAN-A
3-5	86.0	0.136	Online-G
3-5	84.4	0.131	Online-B
3-5	84.2	0.096	JDExploreAcad.
6-7	84.3	0.046	Lan-Bridge
6-7	82.5	0.005	Online-Y
8-10	80.7	−0.086	Online-A
8-11	81.0	−0.123	PROMT
8-11	79.5	−0.159	SRPOL
9-12	79.6	−0.203	HuaweiTSC
11-12	79.4	−0.220	eTranslation

English→Croatian

Range	Ave.	Ave. z	System
1	93.7	0.327	HUMAN-A
2-3	92.6	0.264	HUMAN-st.
2-3	92.0	0.232	Online-B
4	91.2	0.155	Lan-Bridge
5-8	88.5	−0.018	Online-A
5-8	87.3	−0.057	HuaweiTSC
5-8	88.5	−0.068	SRPOL
5-8	87.0	−0.094	NiuTrans
9	84.5	−0.333	Online-G
10	82.3	−0.414	Online-Y

English→Ukrainian

Range	Ave.	Ave. z	System
1	87.1	0.319	HUMAN-A
2-4	84.0	0.124	Online-B
2-4	84.3	0.118	Lan-Bridge
2-4	83.5	0.092	Online-G
5-6	82.8	−0.018	Online-A
5-7	82.0	−0.037	HuaweiTSC
6-7	80.5	−0.105	eTranslation
8-9	79.6	−0.185	Online-Y
8-9	79.8	−0.233	ARC-NKUA

English→Livonian

Range	Ave.	Ave. z	System
1	74.4	1.255	HUMAN-A
2	46.2	0.215	TAL-SJTU
3-4	36.9	-0.147	HuaweiTSC
3-4	36.3	-0.175	TartuNLP
5	33.8	-0.262	Liv4ever
6	17.9	-0.853	NiuTrans

Table 13: Official results of WMT22 General Translation Task for translation out of English or without English. Systems ordered by DA score z-score; systems within a cluster are considered tied; lines indicate clusters according to Wilcoxon rank-sum test $p < 0.05$; rank ranges indicate the number of systems a system significantly underperforms or outperforms; grayed entry indicates resources that fall outside the constraints provided. All language pairs except English→Livonian used document-level evaluation.

guage, or both. There was no distinction of error severity (“major”, “minor” or similar).

All identified errors (issues) were tagged by their possible causes and/or plausible explanations of their origin, as in (Popovic, 2021). Some of the identified “issue types" are equivalent to the typical error classes that can be found in MQM or similar schemes (such as “mistranslation”, “gender”, etc.), while some go beyond that, often including several different intertwining types of errors. Some of them involve single words, while others might involve a large group of words. The main difference between such tags in comparison to MQM or similar tags is that they are related to (linguistically motivated) causes of errors, also taking into account differences between source and target language as well as the translation process, and not only to the “symptoms” manifested in the MT output.

For example, the most frequent issue is related to “rephrasing”, and refers to a sequence of words that is not translated properly for some of the following reasons: 1) the translation of the source words follows the structure of the source language although it should be expressed differently in the target language (rephrasing is needed); 2) rephrasing is needed but incorrectly applied; 3) rephrasing is not needed but is applied, and/or 4) the choice of target words is related to source words but seems random, both in lexical as well as grammatical terms. The issue is manifested by several consecutive different but intertwined types of errors such as case, gender, verb form, mistranslation, function word, omission, addition, word order, etc. Incorrect translation of multi-word expressions and collocations falls under this type.

Overall error rates Table 15 presents the aggregated error rates for each translation, calculated as the number of words which were tagged as any type of error divided by the total number of words in the text. Thus, the interpretation of, for example, the overall error rate of 12.76% for the MT system ONLINE-B is that about 12-13 incorrect words were found in each group of 100 words. The error rates are presented for the entire analysed text, as well as separately for the two domains. The translations are ranked from the lowest to the highest overall error rate.

The ranking is similar to the official direct assessment results presented in Table 13, however there are some different tendencies. The main difference is the preference for human translations – error rates exhibit a clear preference for human translations over MT outputs. While both scores agree on the four best translations (two human and two MT outputs), error rates clearly distinguish the two human translations with about 10% less errors than in the best MT output. Direct assessment scores, however, are all close, ranging from 93.7 to 91.2, and even put student translations at the same rank as the best MT output. The same tendency has been reported in Freitag et al. (2021), where the MQM error annotation on English→German and Chinese→English translations clearly distinguished human translations from MT outputs, contrary to direct assessment scores. These findings indicate that for evaluating human translations in any context (comparing different human translations, comparing with MT outputs), some kind of error annotation should be performed.

Another potentially interesting difference is the system ONLINE-G, which is clearly ranked as second worst by direct assessment, but less clearly as third worst by error annotation. A potential reason is the different nature of errors in different MT systems discussed below. Other differences between the two rankings affect only the mid-range systems which have very close scores in both set-ups.

It can be seen that errors were detected both in human and in machine translations, although the error rates are notably lower in human translations. Overall error rate is lower than 1% for professional translations and lower than 3% for students’ translations, while in MT outputs, the overall error rates range from 12 to 22%.

In human translations, error rates are similar for both domains. In MT outputs, however, the error rates are notably higher for reviews than for news, which is not surprising given that there are much less training resources for reviews. Furthermore, it can be noted that the rankings would be slightly different if only one of the domains were used: NIUTRANS would be ranked higher on news while ONLINE-G would be ranked higher on reviews and HUAWEITSC would be ranked lower. Nevertheless, those variations in rankings can be observed only for the mid-ranged systems where differences in error rates are small anyway.

Comparing machine and human translations Table 16 presents issue types identified in machine and in human translations and their corresponding error rates. In addition, the distribution between the two domains is presented for each, meaning

	SR+DC DA				DA+SQM				MQM	
	Rank	Ave.	Ave. z	Order	Range	Ave.	Ave. z	Order	MQM score	Order
HUMAN-A	-	-	-	-	1-3	82.4	0.137	1	1.223	1
HUMAN-B	1	73.4	0.134	1	8-12	80	-0.029	9	1.997	2
JDExploreAcademy	2	69.8	-0.026	2	3-7	81.5	0.048	6	2.827	6
HuaweiTSC	2	69	-0.034	3	3-7	80.7	0.056	5	3.089	8
AISP-SJTU	2	69.1	-0.063	4	8-10	80.8	-0.013	8	3.187	9
LanguageX	2	69.2	-0.079	5	1-6	82	0.109	2	2.738	5
Online-A	2	69.7	-0.083	6	9-14	79.1	-0.078	10	3.731	11
DLUT	2	68.6	-0.083	7	11-14	79	-0.181	14	-	-
Online-B	2	67.4	-0.089	8	1-3	81.9	0.1	3	2.714	4
Online-G	2	69.9	-0.098	9	3-7	81.4	0.065	4	2.933	7
Online-W	2	66.5	-0.109	10	9-14	78.4	-0.098	12	3.953	12
Lan-Bridge	2	65.3	-0.117	11	4-7	81	0.041	7	2.471	3
Online-Y	2	66.5	-0.122	12	8-12	79.6	-0.086	11	3.281	10
NiuTrans	2	66.3	-0.164	13	11-14	79	-0.107	13	-	-

Table 14: Comparison of three methods of generating human annotations and rankings. Note that each method used different subsets of the test data, and the DA approaches only produced weak clusterings.

en→hr translation		error rate (%) ↓		
		overall	news	reviews
HT	professionals	0.71	0.86	0.60
	students	2.43	2.23	2.59
MT	online-B	12.76	11.19	13.98
	Lan-Bridge	13.42	11.46	14.95
	HuaweiTSC	17.39	12.87	**20.83**
	online-A	17.69	14.30	20.29
	SRPOL	17.96	14.55	20.56
	online-G	18.43	16.60	**19.80**
	NiuTrans	18.99	**13.51**	23.15
	online-Y	21.51	18.48	23.82

Table 15: Percentage of words marked as errors (error rate) in all translations: two human translations (by professional translators and by students) and eight machine translation hypotheses. The percentages are presented for the entire text (overall) and separately for news and for reviews. The translations are ranked from best to worst according to the overall error rate. Bold values indicate domain-specific ranks which are different from the overall rank.

that, for example, 32.2% of all rephrasing errors are found in news and 67.8% in reviews. Issue types are ranked according to their percentage in MT outputs.

The most prominent issues in MT outputs are similar to those reported in in (Popovic, 2021): rephrasing (described at the beginning of the section), ambiguity (different meanings of a word in different contexts), noun phrases (sequences of nouns and possibly adjectives) and omissions (either a part of the source text is omitted or something is missing in the target language), with the error rates ranging from 1% to 5%. Interestingly, the same issue types are the most frequent issues in human translations, too, although with much smaller error rates (less than 0.4%).

The majority of issue types in MT outputs is found more frequently in reviews than in news, although the differences vary. From the most prominent issues, only noun phrase errors are slightly more frequent in news. In human translations, the distribution of issue types between the two domains is more even, although the most prominent four are more frequent in reviews.

Somewhat surprisingly, hallucination errors were identified in the human translation of news. Further manual inspection revealed that in one sentence, a phrase not related to any part of the source text indeed appears in the professional translation. The probable reason is a somewhat specific financial term "like-for-like" meaning "financial growth". The source sentence "Drink-led pubs and bars performed by far the strongest with *like-for-likes* up more than restaurants were down." ended up translated as "Drink-led pubs and bars performed by far the strongest, *while pubs and bars selling both drinks and food had* more up than restaurants were down". The translator probably did not recognise the term and assumed that it refers to something similar to the previously mentioned "drink-led pubs and bars", so they added the phrase about 'pubs and bars selling both drinks and food' which were not mentioned whatsoever in the source. Without this hallucination, all error rates (overall, news and reviews) for professional translations presented in Table 15 would be 0.60%.

Comparing MT systems Table 17 presents the most frequent issue types (with error rate greater than 1%, or, in other words, which were found at least once in each 100 words) in each of the eight MT outputs. The outputs are ranked from best to worst according to the overall error rate (Table 15). For each issue type, its overall error rate together with the separated error rates in news and reviews

en→hr	MT outputs			human translations		
	error	% of the issue type		error	% of the issue type	
issue type	rate %	in news	in reviews	rate %	in news	in reviews
rephrasing	5.12	32.2	**67.8**	0.27	47.9	**52.1**
ambiguity	3.38	32.8	**67.2**	0.21	27.0	**73.0**
noun phrase	2.55	**53.6**	46.4	0.14	20.8	**79.2**
omission	1.22	48.0	**52.0**	0.37	46.9	**53.1**
named entity	0.86	47.4	**52.6**	0.05	50.0	50.0
verb form	0.86	31.3	**68.7**	0.06	**80.0**	20.0
gender	0.85	27.3	**72.7**	0.05	0	**100**
pron/det	0.64	12.7	**87.3**	0.02	0	**100**
preposition	0.54	42.0	**58.0**	0.07	**69.2**	30.8
untranslated	0.52	17.0	**83.0**	0.07	15.4	**84.6**
case	0.50	37.9	**62.1**	0.11	**73.7**	26.3
mistranslation	0.48	38.1	**61.9**	0.07	**61.5**	38.5
addition	0.43	14.8	**85.2**	0.01	0	**100**
source	0.34	2.6	**97.4**	0.02	0	**100**
order	0.28	33.7	**66.3**	0.03	**66.7**	33.3
non-existing	0.25	35.6	**64.4**	0.04	0	**100**
passive	0.19	**53.4**	46.6	0.01	0	**100**
number	0.17	24.1	**75.9**	0.01	0	**100**
-ing	0.16	**59.5**	40.5	0.01	0	**100**
rel. phrase	0.09	**66.7**	33.3	0	0	0
POS ambiguity	0.08	3.4	**96.6**	0	0	0
hallucination	0.07	30.8	**69.2**	0.06	**100**	0
negation	0.06	0	**100**	0	0	0
repetition	0.02	43.8	**56.2**	0.01	**100**	0

Table 16: Identified issues in all MT hypotheses and in both HT references: error rate together with the distribution between news and reviews. The issue types are ordered by their percentage in MT hypotheses. Bold values indicate the domain with the higher amount of a particular issue type.

is shown.

First, it can be noted that in the two best-ranked systems, there are three clearly predominant issue types for both domains: rephrasing, ambiguity and noun phrase. These three issue types are predominant in other systems, too, however with higher error rates.

Furthermore, for all systems, rephrasing errors and ambiguity problems are more frequent in reviews, whereas noun phrase errors are more frequent in news. Also in all systems, there are slightly more omissions in news than in reviews.

When looking at lower ranked systems, it can be noted that not only the error rates for the generally most prominent issue types increase, but also more error types emerge: incorrect verb forms, incorrect gender and problems with pronouns or determiners in reviews.

The most interesting system is ONLINE-G: while the rephrasing error rate is only slightly worse than the two best-ranked systems, and ambiguity and noun phrase errors are also not much worse than some of the higher-ranked systems, it is the only system with notable problems with named entities (more than 2%) and mistranslations (more than 1%) in both domains, as well as generating non-existing words in reviews (more than 1%). This specific distribution of error types could be the reason that this system was clearly ranked as the second worst by direct assessment, although it has similar error rate as some other systems.

In the lowest-ranked systems, apart from the higher error rates for all common issue types, the appearance of untranslated words in reviews can be noted in NIUTRANS, and problems with named entities in news in ONLINE-Y.

Apart from the described quantitative analysis, a qualitative inspection of the translation showed, as can be expected, that the MT outputs generally are close to the source language, without divergences. Nevertheless, some very creative and very nice machine translations were found, too.

Comparing human translations Table 18 presents the most frequent issue types (with error rate greater than 0.1%, or in other words, that were found at least once in each 1000 words) in each of the two human translations. The translations are ranked from best to worst according to the overall error rate (Table 15). For each issue type, its overall error rate together with the separated error rates in news and reviews is shown.

First, it can be noted that the most frequent error in both human translations in omission, being more frequent in student translations. The second issue

en→hr: MT hypotheses				
	most frequent	error rate ↓		
MT system	issue types	overall	news	reviews
online-B	rephrasing	4.14	2.87	5.13
	ambiguity	2.52	1.95	2.96
	noun phrase	1.86	2.77	1.15
Lan-Bridge	rephrasing	4.33	2.98	5.38
	ambiguity	2.62	2.03	3.08
	noun phrase	2.01	2.90	1.31
HuaweiTSC	rephrasing	5.49	3.97	6.65
	ambiguity	3.43	2.43	4.19
	noun phrase	2.64	2.77	2.54
	omission	1.04	1.23	<1
	verb form	<1	<1	1.04
	gender	<1	<1	1.10
online-A	rephrasing	5.06	3.68	6.12
	ambiguity	3.85	2.99	4.50
	noun phrase	2.88	3.20	2.63
	omission	1.11	1.38	<1
	gender	<1	<1	1.18
SRPOL	rephrasing	5.33	4.12	6.25
	ambiguity	3.82	2.69	4.68
	noun phrase	2.69	3.25	2.26
	omission	1.44	1.52	1.38
	verb form	<1	<1	1.00
	pron/det	<1	<1	1.08
online-G	rephrasing	4.59	3.54	5.38
	ambiguity	3.06	2.33	3.61
	noun phrase	2.17	3.28	1.33
	named entity	2.11	2.59	1.75
	omission	1.41	1.61	1.26
	mistranslation	1.37	1.11	1.57
	non-existing	1.06	<1	1.32
	verb form	<1	<1	1.22
	gender	<1	<1	1.04
	pron/det	<1	<1	1.16
NiuTrans	rephrasing	5.76	4.14	6.99
	ambiguity	3.30	2.29	4.08
	noun phrase	2.84	2.93	2.77
	omission	1.69	1.78	1.63
	gender	1.03	<1	1.45
	verb form	<1	<1	1.24
	untranslated	<1	<1	1.14
	pron/det	<1	<1	1.18
online-Y	rephrasing	6.26	5.08	7.16
	ambiguity	4.43	3.75	4.95
	noun phrase	3.29	4.07	2.70
	omission	1.32	1.49	1.20
	verb form	1.15	<1	1.32
	named entity	1.14	1.38	<1
	gender	1.13	<1	1.42
	pron/det	<1	<1	1.18

Table 17: The most frequent issue types (error rate ≥ 1%) in each of the eight MT hypotheses separately, overall as well as separately for news and reviews. The hypotheses are ranked from best to worst according to the overall error rate (Table 15).

en→hr: human translations				
	most frequent	error rate ↓		
	issue types	overall	news	reviews
prof.	omission	0.20	0.13	0.25
	rephrasing	0.14	0.18	0.10
	hallucination	0.11	0.26	0
stud.	omission	0.54	0.64	0.45
	rephrasing	0.41	0.41	0.41
	ambiguity	0.37	0.23	0.47
	noun phrase	0.25	0.13	0.35
	case	0.14	0.18	0.10
	untranslated	0.14	<0.1	0.21
	mistranslation	0.13	0.15	0.10
	preposition	0.11	0.15	<0.1
	verb form	<0.1	0.18	<0.1
	order	<0.1	0.10	<0.1
	named entity	<0.1	0.10	<0.1
	non-existing	<0.1	0	0.14
	gender	<0.1	<0.1	0.14

Table 18: The most frequent issue types (error rate ≥ 0.1%) in each of the two human reference translations separately, overall as well as separately for news and reviews. The translations are ranked from best to worst according to the overall error rate (Table 15).

type is rephrasing, also more frequent in student translations. The third ranked issue in professional translations are hallucinations, which is discussed in one of the previous paragraphs. For students, the third ranked issue are ambiguous words, apparently more problematic in reviews.

Furthermore, a number of issue types with error rate larger than 0.1% in student translations are less frequent or even not appearing at all in professional translations.

Apart from the described quantitative analysis, a qualitative inspection of the translation showed that students generally diverged more from the source language than professionals, which is the opposite of what could be intuitively expected. This is the probable reason that for all MT outputs, both automatic metrics, COMET and CHRF, are lower when calculated using student references.

8 Conclusions

The General Machine Translation Task at WMT 2022 covered 21 translation pairs, 15 of which had English on the source or target side and 6 were without English. Direct assessment (DA) was the main golden truth, although the style varied across language pairs. Into-English translation was evaluated against human reference translation, preserving the order of sentences in a document but not presenting the whole document at once (SR+DC). Out-of-English and non-English pairs offered the context to the annotators and allowed them to revisit the scores assigned to individual segments (DA+SQM), evaluating against the source.

9 Limitations

We opened a research question of testing general capabilities of MT systems. However, we have simplified this approach. Firstly, we only used four domains that are not specialized. Secondly, we used only cleaner sentences avoiding noisy in the source sentences.

Although we accept human judgement as a gold standard, giving us more reliable signal than automatic metrics, we should mention that human annotations are noisy (Wei and Jia, 2021) and their performance is affected by quality of other evaluated systems (Mathur et al., 2020). Moreover, reference-based human judgements are biased by the quality of references.

The error analysis of Croatian translations was carried out by one evaluator. Also, the selected sample is different than the one used for direct assessment.

10 Ethical consideration

Several of the domains contained texts that included personal data, for example the conversational data (See Section 2.5 for more details). Entities were replaced by anonymisation tags (e.g. #NAME#, #EMAIL#) to preserve the anonymity of the users behind the content.

The sentences in Ukrainian datasets (as described in Section 2.4) were collected with users' opt-in consent and any personal data related to people other than well-known people was pseudonymized (using random first names and surnames). Sentences where such pseudonymization would not be enough to preserve reasonable anonymity of the users (e.g. describing events uniquely identifying the persons involved) were not included in the test set.

As described in Section 2.2 and in the linguistic brief (Appendix Section C), inappropriate, controversial and/or explicit content was filtered out prior to translation, particularly keeping in mind the translators and not exposing them to such content or obliging them to translate it. A few sentences containing explicit content managed to escape the filter, and we removed these sentences from the test sets without translation.

Human evaluation using Appraise for collecting human judgements was fully anonymous. Automatically generated accounts associated with annotation tasks with single-sign-on URLs were distributed randomly among pools of annotators and did not allow for storing personal information. For language pairs for which we used calibration HITs, we received lists of tasks completed by an individual anonymous annotator.

Acknowledgments

This task would not have been possible without the sponsorship of monolingual data, test sets translation and evaluation from our partners. Namely Microsoft, Charles University, Toloka, NTT Resonant, LinguaCustodia, Webinterpret, Google, CyberAgent, and Phrase. This work was supported by the European Commission via its H2020 Program (project WELCOME, contract no. 870930) and by 20-16819X (LUSyD). We greatly appreciate the help of Jaroslava Hlaváčová, Lucie Poláková, Pavel Pecina, and Mariia Anisimova with preparation of the Ukrainian↔Czech testset. Additionally, we would like to thank Loïc Barrault, Markus Freitag, Jesús Gonzáles Rubio, Marcin Junczys-Dowmunt, Raheel Qader, Matiss Rikters and many others.

Rachel Bawden's participation was funded by her chair position in the PRAIRIE institute funded by the French national agency ANR as part of the "Investissements d'avenir" programme under the reference ANR-19-P3IA-0001.

Maja Popović's participation was funded by the ADAPT SFI Centre for Digital Media Technology, funded by Science Foundation Ireland through the SFI Research Centres Programme and co-funded under the European Regional Development Fund (ERDF) through Grant 13/RC/2106.

Michal Novák's and Martin Popel's participation was funded by LM2018101 (LINDAT/CLARIAH-CZ) of the Ministry of Education, Youth, and Sports of the Czech Republic.

Yvette Graham's participation was supported by the ADAPT Centre for Digital Content Technology (www.adaptcentre.ie) at Trinity College Dublin funded under the SFI Research Centres Programme (Grant 13/RC/2106) co-funded under the European Regional Development Fund.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Jesujoba Alabi, Lydia Nishimwe, Benjamin Muller, Camille Rey, Benoît Sagot, and Rachel Bawden. 2022. Inria-almanach at wmt 2022: Does transcription help cross-script machine translation? In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Antonios Anastasopoulos, Loïc Barrault, Luisa Bentivogli, Marcely Zanon Boito, Ondřej Bojar, Roldano Cattoni, Anna Currey, Georgiana Dinu, Kevin Duh, Maha Elbayad, Clara Emmanuel, Yannick Estève, Marcello Federico, Christian Federmann, Souhir Gahbiche, Hongyu Gong, Roman Grundkiewicz, Barry Haddow, Benjamin Hsu, Dávid Javorský, Věra Kloudová, Surafel Lakew, Xutai Ma, Prashant Mathur, Paul McNamee, Kenton Murray, Maria Nădejde, Satoshi Nakamura, Matteo Negri, Jan Niehues, Xing Niu, John Ortega, Juan Pino, Elizabeth Salesky, Jiatong Shi, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Yogesh Virkar, Alexander Waibel, Changhan Wang, and Shinji Watanabe. 2022. Findings of the IWSLT 2022 evaluation campaign. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 98–157, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

George Awad, Asad Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, Andrew Delgado, Jesse Zhang, Eliot Godard, Luca Diduch, Alan F. Smeaton, Yvette Graham, and Wessel Kraaij. 2019. Trecvid 2019: An evaluation campaign to benchmark video activity detection, video captioning and matching, and video search & retrieval. In *Proceedings of TRECVID*, volume 2019.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference* *on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. (meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic. Association for Computational Linguistics.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2008. Further meta-evaluation of machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 70–106, Columbus, Ohio. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar Zaidan. 2010. Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics-MATR*, pages 17–53, Uppsala, Sweden. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder. 2009. Findings of the 2009 Workshop on Statistical Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 1–28, Athens, Greece. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. 2011. Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland. Association for Computational Linguistics.

Sheila Castilho. 2020. On the same page? comparing inter-annotator agreement in sentence and document level human machine translation evaluation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1150–1159, Online. Association for Computational Linguistics.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy. European Association for Machine Translation.

Colin Cherry and George Foster. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*.

Hiroyuki Deguchi, Kenji Imamura, Masahiro Kaneko, Yuto Nishida, Yusuke Sakai, Justin Vasselli, Huy-Hien Vu, and Taro Watanabe. 2022. Naist-nict-tit wmt22 general mt task submission. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Liang Ding, Di Wu, and Dacheng Tao. 2021. Improving neural machine translation by bidirectional training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3278–3284, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Adam Dobrowolski, Mateusz Klimaszewski, Adam Myśliwy, Marcin Szymański, Jakub Kowalski, Kornelia Szypuła, Paweł Przewłocki, and Paweł Przybysz. 2022. Samsung r&d institute poland participation in wmt 2022. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Michael Auli, and Armand Joulin. 2021a. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021b. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Ana C. Farinha, M. Amin Farajian, Marianna Buchicchio, Patrick Fernandes, José G. C. de Souza, Helena Moniz, and André F. T. Martins. 2022. Findings of the wmt 2022 shared task on chat translation. In *Proceedings of the Seventh Conference on Machine Translation*. Association for Computational Linguistics.

Christian Federmann. 2018. Appraise evaluation framework for machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 86–88, Santa Fe, New Mexico. Association for Computational Linguistics.

Markus Freitag, Isaac Caswell, and Scott Roy. 2019. APE at scale and its implications on MT evaluation biases. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 34–44, Florence, Italy. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, Ricardo Rei, Nitika Mathur, George Foster, Chi kiu Lo, Craig Stewart, Tom Kocmi, Eleftherios Avramidis, Alon Lavie, and André F. T. Martins. 2022. Results of the wmt22 metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation*. Association for Computational Linguistics.

Thamme Gowda, Zhao Zhang, Chris Mattmann, and Jonathan May. 2021. Many-to-English machine translation tools, data, and pretrained models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 306–316, Online. Association for Computational Linguistics.

Yvette Graham, George Awad, and Alan Smeaton. 2018. Evaluation of automatic video captioning using direct assessment. *PLOS ONE*, 13(9):1–20.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2014. Is machine translation getting better over time? In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 443–451, Gothenburg, Sweden. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2016. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, pages 1–28.

Yvette Graham, Barry Haddow, and Philipp Koehn. 2020. Statistical power and translationese in machine translation evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 72–81, Online. Association for Computational Linguistics.

Bing Han, Yangjian Wu, Gang Hu, and Qiulin Chen. 2022. Lan-bridge mt's participation in the wmt 2022 general translation shared task. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Zhiwei He, Xing Wang, Zhaopeng Tu, Shuming Shi, and Rui Wang. 2022. Tencent ai lab - shanghai jiao tong university low-resource translation system for the wmt22 translation task. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Chang Jin, Tingxun Shi, Zhengshan Xue, and Xiaodong Lin. 2022. Manifold's english-chinese system at wmt22 general mt task. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Josef Jon, Martin Popel, and Ondřej Bojar. 2022. Cunibergamot submission at wmt22 general translation task. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Shivam Kalkar, Yoko Matsuzaki, and Ben LI. 2022. Kyb general machine translation systems for wmt22. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *International Conference on Learning Representations (ICLR)*.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.

Tom Kocmi, Martin Popel, and Ondrej Bojar. 2020. Announcing czeng 2.0 parallel corpus with over 2 gigawords. *CoRR*, abs/2007.03006.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

Ekaterina Lapshinova-Koltunski, Maja Popović, and Maarit Koponen. 2022. DiHuTra: a parallel corpus to analyse differences between human translations. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 337–338, Ghent, Belgium. European Association for Machine Translation.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796, Brussels, Belgium. Association for Computational Linguistics.

Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 2020. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4678–4699, Online. Association for Computational Linguistics.

Guangfeng Liu, Qinpei Zhu, Xingyu Chen, Renjie Feng, Jianxin Ren, Renshou Wu, Qingliang Miao, Rui Wang, and Kai Yu. 2022. The aisp-sjtu translation system for wmt 2022. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Samuel Läubli, Sheila Castilho, Graham Neubig, Rico Sennrich, Qinlan Shen, and Antonio Toral. 2020. A set of recommendations for assessing human–Machine parity in language translation. *Journal of Artificial Intelligence Research (JAIR)*, 67.

Marilena Malli and George Tambouratzis. 2022. Evaluating corpus cleanup methods in the wmt'22 news translation task. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, Emily Pitler, and Leo Wanner. 2018. The first multilingual surface realisation shared task (SR'18): Overview and evaluation results. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 1–12, Melbourne, Australia. Association for Computational Linguistics.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, and Leo Wanner. 2019. The second multilingual surface realisation shared task (SR'19): Overview and evaluation results. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 1–17, Hong Kong, China. Association for Computational Linguistics.

Simon Mille, Anya Belz, Bernd Bohnet, Thiago Castro Ferreira, Yvette Graham, and Leo Wanner. 2020. The third multilingual surface realisation shared task (SR'20): Overview and evaluation results. In *Proceedings of the Third Workshop on Multilingual Surface Realisation*, pages 1–20, Barcelona, Spain (Online). Association for Computational Linguistics.

Alexander Molchanov, Vladislav Kovalenko, and Natalia Makhamalkina. 2022. Promt systems for wmt22 general translation task. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Makoto Morishita, Keito Kudo, Yui Oka, Katsuki Chousa, Shun Kiyono, Sho Takase, and Jun Suzuki. 2022. Nt5 at wmt 2022 general translation task. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2020. JParaCrawl: A large scale web-based English-Japanese parallel corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3603–3609, Marseille, France. European Language Resources Association.

Graham Neubig. 2011. The Kyoto free translation task. http://www.phontron.com/kftt.

Mariana Neves, Antonio Jimeno Yepes, Amy Siu, Roland Roller, Philippe Thomas, Maika Vicente Navarro, Lana Yeganova, Dina Wiemann, Giorgio Maria Di Nunzio, Federica Vezzani, Christel Gerardin, Rachel Bawden, Darryl Johan Estrada, Salvador Lima-Lopez, Eulalia Farre-Maduel, Martin Krallinger, Cristian Grozea, and Aurelie Neveol. 2022. Findings of the wmt 2022 biomedical translation shared task: Monolingual clinical case reports. In *Proceedings of the Seventh Conference on Machine Translation*. Association for Computational Linguistics.

Artur Nowakowski, Gabriela Pałka, Kamil Guttmann, and Mikołaj Pokrywka. 2022. Adam mickiewicz university at wmt 2022: Ner-assisted and quality-aware neural machine translation. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Csaba Oravecz, Katina Bontcheva, David Kolovratnìk, Bogomil Kovachev, and Christopher Scott. 2022.

etranslation's submissions to the wmt22 general machine translation task. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Nikita Pavlichenko, Ivan Stelmakh, and Dmitry Ustalov. 2021. CrowdSpeech and Vox DIY: Benchmark Dataset for Crowdsourced Audio Transcription. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.

Martin Popel. 2020. CUNI English-Czech and English-Polish systems in WMT20: Robust document-level training. In *Proceedings of the Fifth Conference on Machine Translation*, pages 269–273, Online. Association for Computational Linguistics.

Martin Popel, Jindřich Libovický, and Jindřich Helcl. 2022. Cuni systems for the wmt 22 czech-ukrainian translation task. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. 2020a. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(4381):1–15.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. 2020b. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(4381):1–15.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Maja Popovic. 2021. On nature and causes of observed MT errors. In *Proceedings of Machine Translation Summit XVIII: Research Track*, pages 163–175, Virtual. Association for Machine Translation in the Americas.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Reid Pryzant, Youngjoo Chung, Dan Jurafsky, and Denny Britz. 2018. JESC: Japanese-English subtitle corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Matīss Rikters, Marili Tomingas, Tuuli Tuisk, Valts Ernštreits, and Mark Fishel. 2022. Machine translation for Livonian: Catering to 20 speakers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 508–514, Dublin, Ireland. Association for Computational Linguistics.

Dimitrios Roussis and Vassilis Papavassiliou. 2022. The arc-nkua submission for the english-ukrainian general machine translation shared task at wmt22. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Roberts Rozis and Raivis Skadiņš. 2017. Tilde MODEL - multilingual open data for EU languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 263–265, Gothenburg, Sweden. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Weiqiao Shan, Zhiquan Cao, Yuchen Han, Siming Wu, Yimin Hu, Jie Wang, Yi Zhang, Hou Baoyu, Hang Cao, Chenghao Gao, Xiaowen Liu, Tong Xiao, Anxiang Ma, and Jingbo Zhu. 2022. The niutrans machine translation systems for wmt22. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Roman Sudarikov, Martin Popel, Ondřej Bojar, Aljoscha Burchardt, and Ondřej Klejch. 2016. Using MT-ComparEval. In *Translation Evaluation: From Fragmented Tools and Data Sets to an Integrated Ecosystem*, pages 76–82.

Maali Tars, Taido Purason, and Andre Tättar. 2022. Teaching unseen low-resource languages to large translation models. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and*

Evaluation (LREC'12), pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Antonio Toral, Sheila Castilho, Ke Hu, and Andy Way. 2018. Attaining the unattainable? reassessing claims of human parity in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 113–123, Brussels, Belgium. Association for Computational Linguistics.

Daimeng Wei, Zhiqiang Rao, Zhanglin Wu, Shaojun Li, Yuanchang Luo, Yuhao Xie, Xiaoyu Chen, Hengchao Shang, Zongyao Li, Zhengzhe Yu, Jinlong Yang, Miaomiao Ma, Lizhi Lei, Hao Yang, and Ying Qin. 2022. Hw-tsc's submissions to the wmt 2022 general machine translation shared task. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Johnny Wei and Robin Jia. 2021. The statistical advantage of automatic NLG metrics at the system level. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6840–6854, Online. Association for Computational Linguistics.

Changtong Zan, Keqin Peng, Liang Ding, Baopu Qiu, Boan Liu, Shwai He, Qingyu Lu, Zheng Zhang, Chuang Liu, Weifeng Liu, Yibing Zhan, and Dacheng Tao. 2022. Vega-mt: The jd explore academy machine translation system for wmt22. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Hui Zeng. 2022. No domain left behind. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, Portorož, Slovenia. European Language Resources Association (ELRA).

Hao Zong and Chao Bei. 2022. Gtcom neural machine translation systems for wmt22. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

A Statistics of training data

This section describes statistics of the training corpora.

ja-en	**Segments**	**en Toks**	**en Types**
JParacrawl-v3	25.74M	682.78M	2.84M
NewsComm-v16	1.84k	45.28k	6.28k
WikiTitles-v3	757.04k	2.02M	281.88k
WikiMatrix	3.90M	72.32M	1.11M
JESC	2.80M	23.90M	161.38k
KFTT	440.29k	11.54M	190.88k
TED	241.74k	4.95M	64.04k
Total	33.88M	797.55M	3.75M
zh-en	**Segments**	**en Toks**	**en Types**
ParaCrawl(bonus)	14.17M	253.78M	1.87M
NewsComm-v16	313.67k	7.98M	76.36k
WikiTitles-v3	921.96k	2.55M	380.23k
UNPC	17.45M	479.54M	939.62k
CCMT			
WikiMatrix	2.60M	58.62M	1.06M
BackTrans News	19.76M	416.57M	1.19M
Total	55.22M	1.22B	4.01M

Table 19: Training data statistics for ja-en and zh-en. Only the English side statistics are reported, which are obtained after running MosesDecoder's `tokenizer.perl`, similar to Table 20.

Corpus Name	Segments	Tokens		Types	
cs-en		**cs**	**en**	**cs**	**en**
Europarl-v10	644.43k	14.95M	17.38M	172.47k	63.27k
ParaCrawl-v9	50.63M	738.33M	805.54M	4.77M	4.53M
CommonCrawl	161.84k	3.53M	3.93M	210.48k	128.39k
NewsCommentary-v16	253.27k	5.67M	6.27M	176.38k	70.77k
WikiTitles-v3	410.94k	985.54k	1.07M	219.38k	186.37k
WikiMatrix	2.09M	34.82M	39.20M	1.07M	798.09k
Tilde Corpus	2.09M	44.03M	47.83M	349.78k	210.28k
CzEng 2.0	60.98M	757.32M	848.02M	3.68M	2.49M
BackTrans News	126.83M	2.35B	2.66B	5.75M	3.84M
Total	244.10M	3.95B	4.42B		
de-en		**de**	**en**	**de**	**en**
Europarl-v10	1.82M	48.10M	50.47M	371.70k	113.91k
ParaCrawl-v9	278.31M	4.63B	4.90B	31.91M	15.99M
NewsCommentary-v16	388.48k	9.92M	9.83M	215.04k	86.50k
CommonCrawl	2.40M	54.68M	58.90M	1.64M	823.89k
WikiTitles-v3	1.47M	3.23M	3.76M	674.95k	573.28k
WikiMatrix	6.23M	114.22M	118.08M	2.86M	1.83M
Tilde Corpus	5.19M	118.11M	120.82M	986.37k	379.92k
Total	295.81M	4.98B	5.26B		
fr-de		**fr**	**de**	**fr**	**de**
Europarl-v10	1.79M	55.33M	47.49M	144.80k	368.53k
ParaCrawl-v9	7.22M	145.20M	123.51M	1.53M	2.37M
CommonCrawl	622.29k	16.59M	14.23M	332.24k	578.30k
WikiTitles-v3	1.01M	2.54M	2.15M	449.70k	503.34k
NewsCommentary-v16	295.65k	9.34M	7.67M	92.30k	185.28k
Tilde Corpus	4.31M	118.15M	96.00M	391.10k	954.49k
WikiMatrix	3.35M	68.26M	59.85M	1.10M	1.85M
Total	18.60M	415.42M	350.90M		
hr-en		**hr**	**en**	**hr**	**en**
ParaCrawl-v9	3.24M	80.75M	90.83M	1.05M	690.15k
Tilde Corpus	745.62k	14.38M	15.49M	196.78k	109.23k
OPUS	85.56M	928.96M	1.06B	5.26M	4.06M
Total	89.55M	1.02B	1.17B		
ru-en		**ru**	**en**	**ru**	**en**
ParaCrawl-(bonus)	5.38M	99.01M	120.02M	1.73M	1.22M
BackTranslation enru	36.77M	799.38M	839.92M	3.78M	1.92M
Yandex Corpus	1.00M	22.26M	24.30M	697.02k	377.83k
CommonCrawl	878.39k	20.61M	21.54M	712.81k	432.62k
UN Parallel Corpus	985.72k	887.11k	893.73k	5.68k	5.54k
WikiTitles-v3	1.19M	3.24M	3.26M	534.43k	457.93k
NewsCommentary-v16	331.51k	8.37M	8.82M	206.54k	82.93k
WikiMatrix	5.20M	94.00M	102.94M	2.24M	1.59M
Tilde Corpus	34.27k	813.70k	855.68k	62.61k	28.93k
Total	51.77M	1.05B	1.12B		
uk-en		**uk**	**en**	**uk**	**en**
ParaCrawl-(bonus)	13.35M	706.98M	721.28M	1.89M	1.26M
WikiMatrix	2.58M	43.76M	49.06M	1.40M	981.85k
Tilde	1.63k	39.93k	41.15k	8.38k	4.70k
ELRC EU Acts	129.94k	3.20M	3.46M	71.46k	33.52k
OPUS Corpus	48.94M	629.35M	704.32M	4.17M	2.89M
Total	65.01M	1.38B	1.48B		
cs-uk		**cs**	**uk**	**cs**	**uk**
WikiMatirx	848.96k	12.30M	12.28M	586.14k	641.72k
OPUS	11.65M	124.21M	125.84M	1.44M	1.68M
ELRC EU Acts	130.00k	2.86M	3.14M	69.58k	71.67k
Total	12.63M	139.38M	141.26M		
liv-en		**liv**	**en**	**liv**	**en**
Total (from OPUS)	0.77k	23.13k	14.21k	2.51k	2.43k
sah-ru		**sah**	**ru**	**sah**	**ru**
Total (from Yakut corpus)	30.15k	199.94k	225.95k	40.60k	40.64k

Table 20: Statistics for parallel training set provided for General/News Translation Task. All numbers are obtained after running MosesDecoder's `tokenizer.perl`. *Tokens* are the total number of words, whereas *Types* are total number of distinct case-insensitive words. Suffixes, k, M, and B, are short for thousands, millions, and billions, respectively.

B Differences in Human Scores

Tables 23–27 show differences in average standardized human scores for all pairs of competing to-English systems for each language pair. The numbers in each of the tables' cells indicate the difference in average standardized human scores for the system in that column and the system in that row.

Because there were so many systems and data conditions the significance of each pairwise comparison needs to be quantified. We applied Wilcoxon rank-sum test to measure the likelihood that such differences could occur simply by chance. In the following tables $\star$ indicates statistical significance at $p < 0.05$, † indicates statistical significance at $p < 0.01$, and ‡ indicates statistical significance at $p < 0.001$, according to Wilcoxon rank-sum test.

Each table contains final rows showing the average score achieved by that system and the rank range according to Wilcoxon rank-sum test ($p < 0.05$). Gray lines separate clusters based on non-overlapping rank ranges.

	ONLINE-W	CUNI-DOCTRANSFORMER	LAN-BRIDGE	ONLINE-B	JDEXPLOREACADEMY	ONLINE-A	CUNI-TRANSFORMER	ONLINE-G	SHOPLINE-PL	ONLINE-Y	HUMAN-	ALMANACH-INRIA
ONLINE-W	-	0.08⋆	0.08‡	0.10‡	0.14‡	0.15‡	0.15‡	0.16‡	0.22‡	0.28‡	0.42‡	0.43‡
CUNI-DOCTRANSFORMER	-0.08	-	0.01	0.02⋆	0.06	0.07†	0.07†	0.08‡	0.14‡	0.20‡	0.35‡	0.36‡
LAN-BRIDGE	-0.08	-0.01	-	0.01	0.05	0.06	0.07	0.08⋆	0.14†	0.20‡	0.34‡	0.35‡
ONLINE-B	-0.10	-0.02	-0.01	-	0.04	0.05	0.05	0.07	0.12†	0.18‡	0.33‡	0.34‡
JDEXPLOREACADEMY	-0.14	-0.06	-0.05	-0.04	-	0.01	0.01	0.02	0.08†	0.14‡	0.29‡	0.30‡
ONLINE-A	-0.15	-0.07	-0.06	-0.05	-0.01	-	0.00	0.01	0.07	0.13‡	0.28‡	0.29‡
CUNI-TRANSFORMER	-0.15	-0.07	-0.07	-0.05	-0.01	0.00	-	0.01	0.07⋆	0.13‡	0.27‡	0.29‡
ONLINE-G	-0.16	-0.08	-0.08	-0.07	-0.02	-0.01	-0.01	-	0.06	0.12‡	0.26‡	0.27‡
SHOPLINE-PL	-0.22	-0.14	-0.14	-0.12	-0.08	-0.07	-0.07	-0.06	-	0.06†	0.20‡	0.21‡
ONLINE-Y	-0.28	-0.20	-0.20	-0.18	-0.14	-0.13	-0.13	-0.12	-0.06	-	0.14‡	0.16‡
HUMAN-	-0.42	-0.35	-0.34	-0.33	-0.29	-0.28	-0.27	-0.26	-0.20	-0.14	-	0.01
ALMANACH-INRIA	-0.43	-0.36	-0.35	-0.34	-0.30	-0.29	-0.29	-0.27	-0.21	-0.16	-0.01	-
score	0.13	0.06	0.05	0.04	-0.00	-0.01	-0.01	-0.03	-0.09	-0.14	-0.29	-0.30
rank	1	2–3	2–7	3–8	2–8	3–9	3–8	4–9	7–9	10	11–12	11–12

Table 21: Head to head comparison for Czech→English systems

	HUMAN-B	JDExploreAcademy	HuaweiTSC	AISP-SJTU	LanguageX	Online-A	DLUT	Online-B	Online-G	Online-W	Lan-Bridge	Online-Y	NiuTrans
HUMAN-B	-	0.16‡	0.17‡	0.20‡	0.21‡	0.22‡	0.22‡	0.22‡	0.23‡	0.24‡	0.25‡	0.26‡	0.30‡
JDExploreAcademy	-0.16	-	0.01	0.04	0.05	0.06⋆	0.06⋆	0.06	0.07†	0.08‡	0.09‡	0.10‡	0.14‡
HuaweiTSC	-0.17	-0.01	-	0.03	0.05	0.05	0.05⋆	0.06	0.06†	0.08‡	0.08‡	0.09‡	0.13‡
AISP-SJTU	-0.20	-0.04	-0.03	-	0.02	0.02	0.02	0.03	0.04	0.05⋆	0.05⋆	0.06†	0.10‡
LanguageX	-0.21	-0.05	-0.05	-0.02	-	0.00	0.00	0.01	0.02	0.03⋆	0.04⋆	0.04†	0.09‡
Online-A	-0.22	-0.06	-0.05	-0.02	0.00	-	0.00	0.01	0.02	0.03⋆	0.03⋆	0.04†	0.08‡
DLUT	-0.22	-0.06	-0.05	-0.02	0.00	0.00	-	0.01	0.02	0.03	0.03	0.04⋆	0.08‡
Online-B	-0.22	-0.06	-0.06	-0.03	-0.01	-0.01	-0.01	-	0.01	0.02⋆	0.03⋆	0.03†	0.08‡
Online-G	-0.23	-0.07	-0.06	-0.04	-0.02	-0.02	-0.02	-0.01	-	0.01	0.02	0.02⋆	0.07‡
Online-W	-0.24	-0.08	-0.08	-0.05	-0.03	-0.03	-0.03	-0.02	-0.01	-	0.01	0.01	0.06⋆
Lan-Bridge	-0.25	-0.09	-0.08	-0.05	-0.04	-0.03	-0.03	-0.03	-0.02	-0.01	-	0.00	0.05⋆
Online-Y	-0.26	-0.10	-0.09	-0.06	-0.04	-0.04	-0.04	-0.03	-0.02	-0.01	0.00	-	0.04
NiuTrans	-0.30	-0.14	-0.13	-0.10	-0.09	-0.08	-0.08	-0.08	-0.07	-0.06	-0.05	-0.04	-
score	0.13	-0.03	-0.03	-0.06	-0.08	-0.08	-0.08	-0.09	-0.10	-0.11	-0.12	-0.12	-0.16
rank	1	2–6	2–7	2–9	2–9	3–9	4–11	2–9	4–11	8–12	8–12	10–13	12–13

Table 22: Head to head comparison for Chinese→English systems

	Lan-Bridge	Online-W	JDExploreAcademy	Online-G	Online-A	HUMAN-	Online-Y	Online-B	LT22	PROMT
Lan-Bridge	-	0.03⋆	0.04†	0.06†	0.07‡	0.09‡	0.09‡	0.10‡	0.13‡	0.13‡
Online-W	-0.03	-	0.02	0.03	0.05	0.06	0.07†	0.07⋆	0.10‡	0.10‡
JDExploreAcademy	-0.04	-0.02	-	0.02	0.03	0.05	0.05†	0.05⋆	0.09‡	0.09†
Online-G	-0.06	-0.03	-0.02	-	0.01	0.03	0.03⋆	0.03	0.07†	0.07⋆
Online-A	-0.07	-0.05	-0.03	-0.01	-	0.02	0.02	0.02	0.06†	0.06
HUMAN-	-0.09	-0.06	-0.05	-0.03	-0.02	-	0.00	0.01	0.04†	0.04⋆
Online-Y	-0.09	-0.07	-0.05	-0.03	-0.02	0.00	-	0.00	0.04	0.04
Online-B	-0.10	-0.07	-0.05	-0.03	-0.02	-0.01	0.00	-	0.03⋆	0.04
LT22	-0.13	-0.10	-0.09	-0.07	-0.06	-0.04	-0.04	-0.03	-	0.00
PROMT	-0.13	-0.10	-0.09	-0.07	-0.06	-0.04	-0.04	-0.04	0.00	-
score	0.00	-0.02	-0.04	-0.06	-0.07	-0.09	-0.09	-0.09	-0.13	-0.13
rank	1	2–6	2–6	2–7	2–9	2–8	5–10	4–9	8–10	6–10

Table 23: Head to head comparison for German→English systems

	JDExploreAcademy	HuaweiTSC	Online-G	Lan-Bridge	Online-Y	SRPOL	Online-B	Online-A	Online-W	ALMAnaCH-Inria
JDExploreAcademy	-	0.01	0.02	0.05⋆	0.05⋆	0.06⋆	0.07†	0.08†	0.09‡	0.29‡
HuaweiTSC	-0.01	-	0.01	0.03⋆	0.04⋆	0.04⋆	0.05†	0.06†	0.08‡	0.28‡
Online-G	-0.02	-0.01	-	0.02	0.03	0.04	0.04	0.05	0.07⋆	0.27‡
Lan-Bridge	-0.05	-0.03	-0.02	-	0.00	0.01	0.02	0.03	0.05⋆	0.25‡
Online-Y	-0.05	-0.04	-0.03	0.00	-	0.01	0.02	0.03	0.04	0.24‡
SRPOL	-0.06	-0.04	-0.04	-0.01	-0.01	-	0.01	0.02	0.04	0.23‡
Online-B	-0.07	-0.05	-0.04	-0.02	-0.02	-0.01	-	0.01	0.03	0.23‡
Online-A	-0.08	-0.06	-0.05	-0.03	-0.03	-0.02	-0.01	-	0.02	0.22‡
Online-W	-0.09	-0.08	-0.07	-0.05	-0.04	-0.04	-0.03	-0.02	-	0.20‡
ALMAnaCH-Inria	-0.29	-0.28	-0.27	-0.25	-0.24	-0.23	-0.23	-0.22	-0.20	-
score	0.06	0.04	0.03	0.01	0.01	-0.00	-0.01	-0.02	-0.04	-0.24
rank	1–3	1–3	1–8	3–8	3–9	3–9	3–9	3–9	5–9	10

Table 24: Head to head comparison for Russian→English systems

	DLUT	NT5	JDExploreAcademy	LanguageX	Online-B	Online-W	Lan-Bridge	Online-G	Online-A	AISP-SJTU	NAIST-NICT-TIT	Online-Y	KYB	AIST
DLUT	-	0.00	0.01	0.02	0.02	0.02	0.05⋆	0.06†	0.06†	0.09‡	0.09‡	0.10‡	0.13‡	1.35‡
NT5	0.00	-	0.01	0.01	0.02	0.02	0.05⋆	0.06⋆	0.06⋆	0.09‡	0.09†	0.10†	0.12‡	1.35‡
JDExploreAcademy	-0.01	-0.01	-	0.00	0.01	0.01	0.04	0.05	0.05	0.08†	0.08⋆	0.09†	0.11‡	1.34‡
LanguageX	-0.02	-0.01	0.00	-	0.01	0.01	0.04	0.05	0.05	0.07†	0.07⋆	0.09†	0.11‡	1.34‡
Online-B	-0.02	-0.02	-0.01	-0.01	-	0.00	0.03	0.04⋆	0.04⋆	0.07†	0.07⋆	0.08†	0.10‡	1.33‡
Online-W	-0.02	-0.02	-0.01	-0.01	0.00	-	0.03	0.04	0.04	0.06†	0.07⋆	0.08†	0.10‡	1.33‡
Lan-Bridge	-0.05	-0.05	-0.04	-0.04	-0.03	-0.03	-	0.01	0.01	0.03	0.04	0.05	0.07†	1.30‡
Online-G	-0.06	-0.06	-0.05	-0.05	-0.04	-0.04	-0.01	-	0.00	0.02	0.03	0.04	0.06†	1.29‡
Online-A	-0.06	-0.06	-0.05	-0.05	-0.04	-0.04	-0.01	0.00	-	0.02	0.03	0.04	0.06†	1.29‡
AISP-SJTU	-0.09	-0.09	-0.08	-0.07	-0.07	-0.06	-0.03	-0.02	-0.02	-	0.00	0.02	0.04	1.27‡
NAIST-NICT-TIT	-0.09	-0.09	-0.08	-0.07	-0.07	-0.07	-0.04	-0.03	-0.03	0.00	-	0.01	0.04⋆	1.26‡
Online-Y	-0.10	-0.10	-0.09	-0.09	-0.08	-0.08	-0.05	-0.04	-0.04	-0.02	-0.01	-	0.02	1.25‡
KYB	-0.13	-0.12	-0.11	-0.11	-0.10	-0.10	-0.07	-0.06	-0.06	-0.04	-0.04	-0.02	-	1.23‡
AIST	-1.35	-1.35	-1.34	-1.34	-1.33	-1.33	-1.30	-1.29	-1.29	-1.27	-1.26	-1.25	-1.23	-
score	0.07	0.07	0.06	0.05	0.05	0.05	0.02	0.01	0.01	-0.02	-0.02	-0.04	-0.06	-1.28
rank	1–6	1–6	1–9	1–9	1–7	1–9	3–12	4–12	4–12	7–13	7–12	7–13	11–13	14

Table 25: Head to head comparison for Japanese→English systems

	TartuNLP	TAL-SJTU	HuaweiTSC	Liv4ever	NiuTrans
TartuNLP	-	0.04	0.06⋆	0.10†	0.37‡
TAL-SJTU	-0.04	-	0.02	0.07	0.33‡
HuaweiTSC	-0.06	-0.02	-	0.04	0.31‡
Liv4ever	-0.10	-0.07	-0.04	-	0.27‡
NiuTrans	-0.37	-0.33	-0.31	-0.27	-
score	0.02	-0.01	-0.04	-0.08	-0.35
rank	1–2	1–4	2–4	2–4	5

Table 26: Head to head comparison for Livonian→English systems

	Lan-Bridge	Online-B	HuaweiTSC	Online-A	PROMT	Online-G	Online-Y	ARC-NKUA	ALMAnaCH-Inria
Lan-Bridge	-	0.00	0.01⋆	0.04⋆	0.06†	0.07†	0.12‡	0.13‡	0.29‡
Online-B	0.00	-	0.01†	0.04⋆	0.06‡	0.07‡	0.12‡	0.13‡	0.29‡
HuaweiTSC	-0.01	-0.01	-	0.03	0.05	0.06	0.11†	0.12†	0.28‡
Online-A	-0.04	-0.04	-0.03	-	0.02	0.03	0.08†	0.09†	0.25‡
PROMT	-0.06	-0.06	-0.05	-0.02	-	0.01	0.06⋆	0.07⋆	0.24‡
Online-G	-0.07	-0.07	-0.06	-0.03	-0.01	-	0.05⋆	0.06⋆	0.22‡
Online-Y	-0.12	-0.12	-0.11	-0.08	-0.06	-0.05	-	0.01	0.17‡
ARC-NKUA	-0.13	-0.13	-0.12	-0.09	-0.07	-0.06	-0.01	-	0.16‡
ALMAnaCH-Inria	-0.29	-0.29	-0.28	-0.25	-0.24	-0.22	-0.17	-0.16	-
score	0.05	0.05	0.04	0.01	-0.01	-0.02	-0.07	-0.08	-0.25
rank	1–2	1–2	3–6	3–6	3–6	3–6	7–8	7–8	9

Table 27: Head to head comparison for Ukrainian→English systems

C Preprocessing cleanup brief for linguists

In this task, we wish to check the data to remove all inappropriate content, remove repetitive content, or correct minor problems with the text.

The data is automatically broken down into individual sentences, which may be wrong sentence splitting. Each document is separated by empty lines. Keep the document-separators intact, split long documents into several by adding empty lines if necessary based on the context (some documents may be merged). In general, documents should be under 30 sentences long.

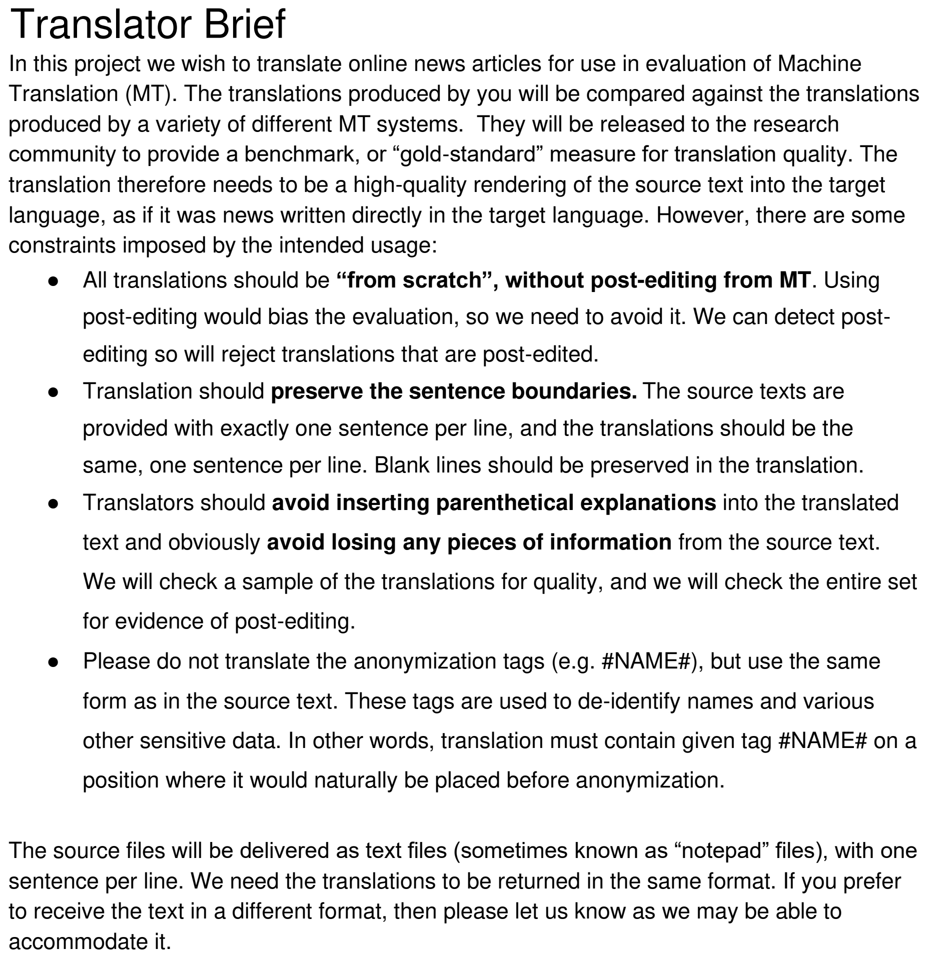
In the first step, check if a document shouldn't be removed (delete sentences from document) based on the following conditions, be on the save side, rather remove documents where you are uncertain. The conditions for removal of documents are as follows:

- Remove inappropriate content (such as sexually explicit, vulgar, or otherwise inappropriate)
- Remove controversial content (propagandist, controversial political topics, etc.)
- Remove content that is too noisy or doesn't resemble natural text (such as documents badly formatted, hard to understand, containing unusual language, lists or other structured data generated automatically)
- Remove repeated/similar content already part of previous documents

For documents that are not removed, do minor corrections (do not try reformulating the content). The main goal is to make sure each line contains a single sentence (or is empty line which represent document boundaries). The result should be documents that are fluent when reading. Here is a non-complete list of phenomena to pay attention to:

- Each line must be a single sentence, remove anything that dangles around or doesn't fit the context. Also reconnect sentences that have been accidentally split (for example trailing words or punctuation should be appended to the previous line).
- You may do small corrections to make the text cleaner (adding punctuation, correcting small typos, etc.). If text would need more correction, remove whole document. Also, do not polish everything.
- Sentences containing a short phrase or single words that are not necessary for the context (like "Description:" or emoticons like ":)") can be removed.

D Translator Brief for General MT



Translator Brief

In this project we wish to translate online news articles for use in evaluation of Machine Translation (MT). The translations produced by you will be compared against the translations produced by a variety of different MT systems. They will be released to the research community to provide a benchmark, or “gold-standard” measure for translation quality. The translation therefore needs to be a high-quality rendering of the source text into the target language, as if it was news written directly in the target language. However, there are some constraints imposed by the intended usage:

- All translations should be **“from scratch”, without post-editing from MT**. Using post-editing would bias the evaluation, so we need to avoid it. We can detect post-editing so will reject translations that are post-edited.
- Translation should **preserve the sentence boundaries.** The source texts are provided with exactly one sentence per line, and the translations should be the same, one sentence per line. Blank lines should be preserved in the translation.
- Translators should **avoid inserting parenthetical explanations** into the translated text and obviously **avoid losing any pieces of information** from the source text. We will check a sample of the translations for quality, and we will check the entire set for evidence of post-editing.
- Please do not translate the anonymization tags (e.g. #NAME#), but use the same form as in the source text. These tags are used to de-identify names and various other sensitive data. In other words, translation must contain given tag #NAME# on a position where it would naturally be placed before anonymization.

The source files will be delivered as text files (sometimes known as “notepad” files), with one sentence per line. We need the translations to be returned in the same format. If you prefer to receive the text in a different format, then please let us know as we may be able to accommodate it.

E Additional statistics of the test sets

Table 28 shows the type-token ratios for the source and target side of each of the test sets, shown for each available domain. As mentioned previously, texts are tokenised using the language-specific Spacy models (Honnibal and Montani, 2017) where available. For Czech, Livonian and Yakut, for which Spacy models are not available, we took as a rough approximation models for Croatian, Finnish and Russian respectively. The type-token ratio is calculated as the number of unique tokens divided by the total number of tokens. The absolute value depends not only on the lexical diversity of the text but also on the morphological complexity of the language in question.

	Type-token ratio (source)					Type-token ratio (target)				
	conversation	ecommerce	news	social	other	conversation	ecommerce	news	social	other
cs-en	-	-	0.40	0.38	-	-	-	0.21	0.22	-
cs-uk	-	-	-	-	0.34	-	-	-	-	0.32
de-en	0.25	0.36	0.35	0.29	-	0.18	0.24	0.26	0.21	-
de-fr	0.25	0.36	0.35	0.29	-	0.20	0.24	0.26	0.23	-
en-cs	0.15	0.24	0.26	0.23	-	0.23	0.36	0.41	0.36	-
en-de	0.15	0.24	0.26	0.23	-	0.15	0.27	0.3	0.26	-
en-hr	-	0.20	0.24	-	-	-	0.31	0.36	-	-
en-ja	0.15	0.24	0.26	0.23	-	0.10	0.17	0.18	0.18	-
en-liv	-	-	-	-	0.25	-	-	-	-	0.34
en-ru	0.15	0.24	0.26	0.23	-	0.21	0.35	0.39	0.33	-
en-uk	0.15	0.24	0.26	0.23	-	0.20	0.34	0.37	0.34	-
en-zh	0.15	0.24	0.26	0.23	-	0.13	0.22	0.26	0.25	-
fr-de	0.19	0.26	0.27	0.26	-	0.18	0.30	0.31	0.28	-
ja-en	0.24	0.20	0.23	0.24	-	0.24	0.24	0.26	0.24	-
liv-en	-	-	-	-	0.34	-	-	-	-	0.25
ru-en	-	0.44	0.35	-	0.43	-	0.26	0.20	-	0.27
ru-sah	-	-	-	-	0.34	-	-	-	-	0.38
sah-ru	-	-	-	-	0.38	-	-	-	-	0.34
uk-cs	-	-	-	-	0.28	-	-	-	-	0.26
uk-en	-	-	-	-	0.28	-	-	-	-	0.13
zh-en	0.24	0.30	0.25	0.27	-	0.17	0.21	0.17	0.20	-

Table 28: Type-token ratio for individual source languages used in the general translation test sets.

F News Task System Submission Summaries

F.1 AISP-SJTU (Liu et al., 2022)

This paper describes AISP-SJTU's participation in WMT 2022 shared general mt task on English->Chinese, Chinese->English, English->Japanese and Japanese->English with constrained training data. Our systems are based on the Transformer architecture with several novel and effective variants, including network depth and internal structure. In our experiments, we employ data filtering, large-scale back-translation, knowledge distillation, forward-translation, iterative in-domain knowledge finetune and model ensemble.

F.2 AIST (no associated paper)

The model was trained similarly to Optimus (Li et al., 2020) with the difference of using BERT (Devlin et al., 2019) for both encoding and decoding instead of BERT for encoding and GPT-2 for decoding as in Optimus, therefore enabling non-autoregressive sequence-to-sequence modeling. We used the pre-trained "bert-base-cased" configuration for English and the "bert-base-japanese" from CL Tohoku for Japanese.

F.3 ALMAnaCH-Inria (Alabi et al., 2022)

ALMAnaCH-Inria's primary submissions are multilingual transformer models between English, Russian, Ukrainian and Russian. The models exploit a dedicated Latin-script transcription convention designed to represent the Slavic languages in a way that maximises character- and word-level correspondences between them as well as with English. For directions where the target language is not English, this involves a final translation step into the original script. Our hypothesis was that bringing the languages closer together could boost vocabulary sharing and have a positive impact on machine translation results. Initial results indicate that the transcription strategy was not successful, resulting in lower results than baselines. We nevertheless submit these models as our primary systems.

F.4 AMU (Nowakowski et al., 2022)

AMU submission is a weighted ensemble of 4 models based on the transformer-big architecture. Models use source factors to utilize the information about named entities present in the input. Each of the models in the ensemble was trained using only the data provided by the shared task organizers. A noisy back-translation technique was used to augment the training corpora. One of the models in the ensemble is a document-level model, trained on parallel and synthetic longer sequences. During the sentence-level decoding process, the ensemble generated the n-best list (n=200). The n-best list was merged with the n-best list (n=50) generated by a single document-level model which translated multiple sentences at a time. Finally, existing quality estimation models and minimum Bayes risk decoding were used to rerank the n-best list so that the best hypothesis is chosen according to the COMET evaluation metric.

F.5 ARC-NKUA (Roussis and Papavassiliou, 2022)

The ARC-NKUA submission to the WMT22 General Machine Translation shared task concerns the unconstrained tracks of the English-Ukrainian and Ukrainian-English translation directions. The 2 Neural Machine Translation systems are based on Transformer models and our primary submissions were determined through experimentation with (a) checkpoint averaging, (b) ensemble decoding, (c) continued training with a subset of the training data, (d) data augmentation with back-translated monolingual data, and (e) post-processing of the translation outputs. We used various techniques to clean and filter the data provided by the organizers, as well as the additional parallel and monolingual data which we acquired from various sources.

F.6 CUNI-Bergamot (Jon et al., 2022)

CUNI-Bergamot submission is based on block-backtranslation method and MBR decoding using neural metrics. Block-BT is a method which switches between blocks of authentic parallel and backtranslated data during training based on a predefined pattern. The paper compares various parameters of the block-BT method: block size, checkpoint averaging methods, using only BT or also forward translation. The authors also show that MBR decoding can profit from more diverse checkpoints created by this method, as opposed to traditional mixed data training.

F.7 CUNI-DocTransformer (Jon et al., 2022)

Exactly the same as submitted in WMT20 (Popel, 2020), document-level Transformer trained with Block Backtranslation.

F.8 CUNI-Transformer (Jon et al., 2022)

The English↔Czech sentence-level models are exactly the same as submitted in WMT20 (Popel, 2020). The Ukrainian↔Czech models are very similar, also trained with Block Backtranslation. The Czech→Ukrainian system uses in addition special preprocessing (romanization of the Ukrainian side and a novel vocabulary-based inline casing on both sides).

F.9 CharlesTranslator (Popel et al., 2022)

Charles Translator for Ukraine is a free Czech-Ukrainian online translation service available for the public at `https://translator.cuni.cz` and as an Android app. It was developed at Charles University in March 2022 to help refugees from Ukraine by narrowing the communication gap between them and other people in the Czech Republic. It is based on Transformer and Block Backtranslation (Popel et al., 2020a).

F.10 DLUT (no associated paper)

We participate in the WMT 2022 general translation task in 2 language pairs and four language directions, English-Chinese and English-Japanese. Our submission use standard Transformer bilingual models.

We mainly improve performance by data filtering, large-scale data generation (i.e., back-translation, forward-translation, knowledge distillation, R2L training), domain finetuning, model ensemble and post-editing.

F.11 GTCOM (Zong and Bei, 2022)

This submission is based on Transformer architecture and involves data augmentation techniques.

F.12 HuaweiTSC (Wei et al., 2022)

This paper describes the submission of huawei translation services center (HW-TSC) to WMT22 general MT translation task.

F.13 JDExploreAcademy (Zan et al., 2022)

We push the limit of our previous work – bidirectional training (Ding et al., 2021) for machine translation by scaling up two main factors, i.e. language pairs and model sizes, namely the Vega-MT system. As for language pairs, we scale the "bidirectional" up to the "multidirectional" settings, covering all competitive high-resource languages, including en-de, en-cs, en-ru, en-zh, and en-ja, to exploit the common knowledge across languages, and transfer them to the downstream bilingual tasks. As for model size, we scale the transformer-big up to the extremely large model that owns nearly 4.7 Billion parameters, to fully enhance the model capacity for our Vega-MT. Also, we adopt the widely-used data augmentation strategies, e.g. back translation, knowledge distillation, cycle translation, and bidirectional self-training to comprehensively exploit the bilingual and monolingual data. To adapt our Vega-MT to the general domain test set, the noisy channel reranking and generalization tuning are employed.

F.14 KYB (Kalkar et al., 2022)

KYB team participated in the WMT22 general machine translation task on English-to-Japanese and Japanese-to-English directions. Our submissions are based on the transformer model with base setting. We employed several techniques to improve system's performance, such as data cleaning and selection, model ensembling/averaging, beam search, fine-tuning, and post-processing.

F.15 LT22 (Malli and Tambouratzis, 2022)

Our submission consists of translations produced from a series of NMT models of the following two language pairs: german-to-english and german-to-french. All the models are trained using only the parallel training data specified by WMT22. The models follow the transformer architecture employing eight attention heads and six layers in both the encoder and decoder. It is also worth mentioning that, in order to limit the computational resources that we would use during the training process, we decided to train the majority of models by limiting the training to 21 epochs. Moreover, the translations submitted at WMT22 have been produced using the test data released by the WMT22. The aim of our experiments has been to evaluate methods for cleaning-up a parallel corpus to determine if this will lead to a translation model producing more accurate translations. For each language pair, the base NMT models have been trained from raw parallel training corpora, while the additional NMT models have been trained with corpora subjected to a special cleaning process with the following tools: Bifixer and Bicleaner. It should be mentioned that the Bicleaner repository doesn't provide pre-trained classifiers for the above language pairs, consequently we trained probabilistic dictionaries in order to produce new models. The fundamental differences between these NMT models produced are mainly related to the quality and the quantity of the training data, while there are very few differences in the training parameters. To complete this work, we used the following three tools:(i) MARIAN NMT (Version: v1.11.5), which was used for the training of the NMT models and (ii) Bifixer and (iii) Bicleaner, which were used in order to correct and clean the parallel training data. Concerning the Bifixer and Bicleaner tools, we followed all the steps as described meticulously in the relevant article.

F.16 Lan-Bridge (Han et al., 2022)

Team Lan-Bridge's submission are transformer base models. For non-Chinese language pairs, we trained some multilingual models. For Chinese-English and English-Chinese, we train seperated models for each direction.

F.17 LanguageX (Zeng, 2022)

LanguageX submission is an ensemble model equipped with our recent technique of fast domain adaptation and data selection.

F.18 Liv4ever (Rikters et al., 2022)

The submitted translations were generated by an ensemble of three different iterations of multi-lingual transformer models trained on Latvian, Estonian, English and Livonian data from the constrained track. All parallel data were filtered (?) before training. After initial training the models were further improved by performing iterative back-translation of batches of 200,000 sentences from each language to the other languages (Livonian monolingual data was upscaled) for four iterations. The ensemble was composed of the single best checkpoint from the last three iterations of the back-translation process.

F.19 NAIST-NICT-TIT (Deguchi et al., 2022)

This paper describes the NAIST-NICT-TIT submission to the WMT22 general machine translation task. We participated in this task in the English-Japanese language pair. Our system is built on an ensemble of Transformer big models, k-nearest-neighbor machine translation (kNN-MT) (Khandelwal et al., 2021), and reranking.

Our base translation system is a combination of kNN-MT and an ensemble of four Transformer big models. Each of the Transformer model instances is trained using a different random seed, and we reuse one of the models for kNN-MT. A notable point of our system is that we construct the datastore for kNN-MT from back-translated monolingual data. We find that using the back-translated data improves translation performance when compared to using a parallel training corpus for the datastore.

We designed a reranking system to select a sentence from among the n-best sentences generated by the base translation system. For each translation hypothesis, the reranker computes a weighted sum of multiple model scores. It then selects the hypothesis with the highest score. We used k-best batch MIRA (Cherry and Foster) to select the weights for the model scores that maximize the BLEU score of the development set. We use context-aware model scores to improve the document-level consistency of the translation.

F.20 NT5 (Morishita et al., 2022)

The NT5 team submission is standard ensemble Transformer models equipped with several extensions, including our recent techniques, followed by a reranking module based on source-to-target, target-to-source, and masked language models. We also applied data augmentation and selection techniques to training data of the Transformer models.

F.21 NiuTrans (Shan et al., 2022)

This paper describes NiuTrans neural machine translation systems of the WMT22 General MT task with constrained data sets. We participated in Chinese to English, English to Croatian, and Livonian-English total of three tasks. We mainly utilized iterative back-translation, iterative knowledge distillation, and iterative fine-tuning. We also use various Transformer variants to improve the model's performance further, e.g., ODE-Transformer, UMST. Moreover, we tried some multi-domain methods, such as multi-domain model structure and multi-domain data clustering method, to adapt to this year's multi-domain test set. We also tried some methods to build a machine translation system using pre-trained language models.

F.22 OpenNMT (no associated paper)

In this paper, we first benchmark the mainstream translators on the English-to-German task by making sure we take into account: - The changes that occurred in the WMT test sets starting 2019 - The post-processing differences between systems - The recent research in automatic metrics beyond BLEU Over the past 3 years, WMT has shown that both OnlineW and FacebookAI have a clear lead in the human evaluations. When looking at various metrics, we make the assumptions that one reason comes from the very good fluency which exposes a low perplexity when measuring with a GPT-2 language model.

We will therefore try 3 types of experiments: 1) filter various datasets with a GPT-2 model to retain only sentences under a given threshold. 2) Use a noisy channel decoding reranking method (used by FacebookAI) and maybe by OnlineW since their API is way slower then G/M/A. 3) Use a GPT-2 large model distillation during NMT training.

Given the training time of the last experiment we were not able to submit this system, however we will continue and report results in the paper.

F.23 PROMT (Molchanov et al., 2022)

The PROMT systems are trained with the MarianNMT toolkit. All systems use the transformer-big configuration. We use BPE for text encoding, the vocabulary sizes vary from 24k to 32k for different language pairs. All systems are unconstrained. We use all data provided by the WMT organizers, all publicly available data and some private data.

F.24 SRPOL (Dobrowolski et al., 2022)

We present the work of Samsung R&D Institute Poland in WMT 2022 General MT solution for medium to low resource languages: Russian and Croatian. Our approach combines iterative back-translation with noise and iterative distillation. We investigated different monolingual resources and compared their effects on the final translation. We used available BERT-like models to classify texts and to distinguish text domains. We attempted to predict ensemble weight vectors based on BERT-like domain classification for individual sentences. The final models achieved quality comparable to the best online translators using only limited resources during training.

F.25 TAL-SJTU (He et al., 2022)

TAL-SJTU submission is based on M2M100 (Fan et al., 2021a) with novel techniques that adapt it to the target language pair: (1) We propose a cross-model word embedding alignment method that transfers a pre-trained word embedding to M2M100, enabling it to support Livonian. (2) We also utilize Estonian and Latvian languages as auxiliary languages for training and pivot languages for data augmentation. (3) Finally, the best result was achieved after fine-tuning the model using the validation set and online back-translation. In model evaluation: (1) We find that previous work (Rikters et al., 2022) underestimated the translation performance of Livonian due to inconsistency in Unicode normalization, which may cause a discrepancy of up to 19 BLEU score. (2) In addition to the standard validation set, we also employ round-trip BLEU to evaluate the models, which we find a more appropriate way for this task.

F.26 TartuNLP (Tars et al., 2022)

TartuNLP's submission is a model based on Transformers. Our main approach was utilizing large pre-trained multilingual neural machine translation models, specifically the M2M-100 model (Fan et al., 2021b). In our systems we used the 1.2 billion parameter model. We fine-tuned the pre-trained model (more specifically we performed cross-lingual transfer learning) to our data, which consisted of WMT22 liv-en, en-liv data and other data from the Finno-Ugric language family for support. The main pipeline was the following: fine-tuning with original parallel data, then two iterations of back-translation and finally fine-tuning on original parallel data again.

F.27 eTranslation (Oravecz et al., 2022)

eTranslations's Fr-De system is an ensemble of 4 big transformers, trained from all available parallel data and with additional tagged, back-translated data generated from a 30M subset of various German monolingual corpora. The monolingual and original parallel data is cleaned up and filtered with heuristic rules. In the model trainings, the original parallel data is upsampled to a 1:1 ratio. Each transformer model is then fine tuned for 3 epochs on the original parallel data. The models use a 32k SentencePiece vocabulary. The SentencePiece module as built in the Marian toolkit is used for end-to-end text processing, without the standard pre- and postprocessing steps of truecasing, or (de)tokenization.

The En-Uk system is an ensemble of 4 multilingual (En -> Uk, Ru) big transformers, trained from all available parallel data. Each transformer model is then fine tuned only on the En-Uk data for about 50 epochs and the best checkpoint is used in the ensemble. Vocabulary and pre/postprocessing settings are the same as the Fr-De system. The En-Ru system is built with the same setup as the En-Uk, except it is an ensemble of 3 models.

F.28 manifold (Jin et al., 2022)

Manifold's English-Chinese System at WMT22 is an ensemble of 4 models, each trained by one of four different configurations and fine-tuned by applying scheduled-sampling. The four configurations are DeepBig (Xenc), DeepLarger (Xenc), DeepBigTalkingHeads (Xenc) and DeepBig (LaBSE). DeepBig is an extension to TransformerBig, the only difference is the former has 24 encoder layers. DeepLarger has 20 encoder layers and its FFN dimension is 8192. *TalkingHead applies talking-heads trick. For Xenc configs, we selected monolingual and parallel data that is similar to the past newstest datasets using Xenc, and for LaBSE, we cleaned the officially provided parallel data using LaBSE pretrained model.

F.29 shopline-pl

The model we submitted is based on the query results of the transformer and its variants, which includes the integration effect of different models and incorporates the reserved word mechanism.

G Automatic scores

This section contains automatic metric scores. While human judgement is the official ranking of systems and their performance, we share automatic scores to show expected system performance for various testsets.

We use COMET (Rei et al., 2020) as the primary metric and ChrF (Popović, 2015) as the secondary metric, following recommendation by (Kocmi et al., 2021). We present BLEU (Papineni et al., 2002) scores as it is still widely used metric. The COMET scores are calculated with the default model `wmt20-comet-da`. The ChrF and BLEU scores are calculated using SacreBLEU with signature (Post, 2018) is `chrF2|nrefs:all|case:mixed|eff:yes|nc:6|nw:0|space:no|version:2.0.0`. Scores are multiplied by 100.

The different suffix represents the name of reference used for calculation (A, B, C, stud), references has been translated by different translators but with the same sponsor. A notable difference is Czech-English, where we are missing reference "A" for it's low quality, which was partly corrected and placed under "C". The second exception is Croatian reference "stud" which was created by students in contrast to "A" prepared by professionals. Lastly, testsets liv-en and ru-sah are reverse testsets to their opposite counterparts (i. e. "en" and "sah" are original sources)

Table 29: Automatic metric scores for en-cs.

System	COMET$_B$ ↑	COMET$_C$	ChrF$_B$	ChrF$_C$	BLEU$_B$	BLEU$_C$
Online-W	97.8	79.3	68.2	51.8	45.8	25.0
Online-B	97.5	76.6	69.0	52.7	48.2	27.0
CUNI-Bergamot	96.0	79.0	63.2	50.3	38.6	24.4
JDExploreAcademy	95.3	77.8	65.1	51.8	41.4	25.5
Lan-Bridge	94.7	73.8	68.2	52.3	45.6	25.9
Online-A	92.2	71.1	65.8	50.8	41.8	24.5
CUNI-DocTransformer	91.7	72.2	63.9	50.8	39.8	25.2
CUNI-Transformer	86.6	68.6	62.1	50.1	37.7	24.5
Online-Y	83.7	62.3	62.9	49.0	37.8	22.8
Online-G	82.3	61.5	62.8	49.0	38.1	22.7

Table 30: Automatic metric scores for en-de.

System	COMET$_A$ ↑	COMET$_B$	ChrF$_A$	ChrF$_B$	BLEU$_A$	BLEU$_B$
Online-W	65.5	64.4	64.1	62.7	36.6	35.3
JDExploreAcademy	63.2	62.5	64.3	63.8	37.8	38.2
Online-B	62.3	61.9	64.6	64.1	38.4	38.3
Online-Y	61.1	60.9	63.7	63.5	37.0	37.2
Online-A	60.6	60.0	63.9	63.6	36.5	37.2
Online-G	60.2	59.3	63.4	63.1	36.4	36.6
Lan-Bridge	58.8	58.3	64.1	63.7	36.1	36.5
OpenNMT	57.2	57.0	62.1	61.5	35.7	35.7
PROMT	55.8	55.3	62.8	62.2	36.1	36.0

Table 31: Automatic metric scores for en-hr.

System	COMET$_A$ ↑	COMET$_{stud}$	ChrF$_A$	ChrF$_{stud}$	BLEU$_A$	BLEU$_{stud}$
Online-B	80.4	77.6	58.5	57.6	31.5	29.8
Lan-Bridge	79.6	76.7	58.5	57.4	31.5	29.7
GTCOM	77.4	74.7	58.1	57.0	30.7	28.6
Online-A	69.5	67.1	56.5	55.9	29.1	28.1
SRPOL	69.4	67.6	56.3	55.6	29.1	27.8
HuaweiTSC	67.6	66.3	56.8	56.1	29.9	28.6
NiuTrans	65.5	63.4	56.3	55.6	29.3	28.1
Online-G	64.2	63.0	53.2	52.5	25.7	24.3
Online-Y	56.7	55.1	54.3	53.6	26.6	25.1

Table 32: Automatic metric scores for en-ja.

System	COMET_A ↑	ChrF_A	BLEU_A
JDExploreAcademy	65.1	36.1	41.5
NT5	64.1	36.8	42.5
LanguageX	62.1	36.1	41.7
Online-B	60.8	35.5	41.2
DLUT	60.5	36.1	41.8
Online-W	59.8	35.2	40.8
Online-Y	56.8	34.4	39.9
Lan-Bridge	56.5	34.1	39.4
Online-A	53.6	34.1	38.8
NAIST-NICT-TIT	53.3	33.8	39.2
AISP-SJTU	52.4	33.9	39.3
KYB	31.8	28.6	33.1
Online-G	24.9	28.0	32.1

Table 33: Automatic metric scores for en-liv.

System	COMET_A ↑	ChrF_A	BLEU_A
TAL-SJTU	-29.5	43.8	17.0
TartuNLP	-36.8	39.2	15.0
HuaweiTSC	-38.9	37.7	12.8
Liv4ever	-39.4	39.6	14.7
NiuTrans	-81.9	30.5	12.3

Table 34: Automatic metric scores for en-ru.

System	COMET_A ↑	ChrF_A	BLEU_A
Online-W	75.1	58.3	32.4
Online-G	73.1	59.5	32.8
Online-B	72.9	59.7	34.9
Online-Y	69.8	58.3	33.2
JDExploreAcademy	69.6	58.4	32.7
Lan-Bridge	67.3	59.0	32.6
Online-A	67.3	58.1	33.1
PROMT	60.3	56.1	30.6
SRPOL	59.7	56.4	30.4
HuaweiTSC	59.2	56.1	30.8
eTranslation	57.9	55.8	29.8

Table 35: Automatic metric scores for en-uk.

System	COMET_A ↑	ChrF_A	BLEU_A
Online-B	73.2	59.3	32.5
GTCOM	72.0	59.0	30.8
Online-G	69.9	57.2	27.2
Lan-Bridge	65.7	58.8	29.5
Online-A	60.9	56.0	28.0
eTranslation	54.5	54.8	26.2
HuaweiTSC	54.4	54.8	26.5
Online-Y	51.9	54.9	26.9
ARC-NKUA	49.2	54.0	25.2

Table 36: Automatic metric scores for en-zh.

System	COMET$_A$ ↑	COMET$_B$	ChrF$_A$	ChrF$_B$	BLEU$_A$	BLEU$_B$
GTCOM	64.7	69.4	44.1	45.7	47.7	50.5
LanguageX	63.8	71.5	49.1	53.1	54.3	59.8
Online-B	61.8	80.4	44.4	68.6	49.1	73.7
JDExploreAcademy	61.7	70.6	44.6	51.1	49.7	57.6
Lan-Bridge	61.4	69.4	42.8	49.2	48.3	56.0
Online-W	61.0	69.5	41.1	47.7	44.8	52.6
Manifold	60.1	71.2	44.2	54.3	48.7	59.6
Online-Y	59.7	71.7	42.3	54.0	46.8	59.9
HuaweiTSC	59.5	73.1	44.5	58.1	49.7	64.4
Online-A	57.3	70.1	42.5	55.5	46.4	60.7
AISP-SJTU	56.5	66.6	43.9	50.9	48.8	57.3
DLUT	52.1	63.0	41.3	50.1	45.2	55.4
Online-G	51.2	62.5	39.4	49.8	43.9	55.2

Table 37: Automatic metric scores for cs-en.

System	COMET$_B$ ↑	COMET$_C$	ChrF$_B$	ChrF$_C$	BLEU$_B$	BLEU$_C$
Online-W	77.5	45.6	79.3	52.0	64.2	23.8
JDExploreAcademy	74.7	49.0	74.4	53.7	54.9	25.1
Lan-Bridge	71.8	47.2	74.0	54.0	54.5	25.5
Online-B	71.8	47.4	73.8	54.0	54.3	25.5
CUNI-DocTransformer	70.6	45.3	72.2	53.0	51.9	24.8
Online-A	69.8	44.3	73.4	53.4	53.3	25.0
CUNI-Transformer	69.2	43.2	71.7	52.0	51.6	23.9
Online-G	63.0	38.8	70.3	52.1	48.5	23.0
SHOPLINE-PL	61.1	39.6	69.2	53.2	46.8	24.6
Online-Y	58.6	35.2	67.9	51.5	44.6	23.1
ALMAnaCH-Inria	19.3	4.9	56.9	48.3	29.9	19.7

Table 38: Automatic metric scores for de-en.

System	COMET$_A$ ↑	COMET$_B$	ChrF$_A$	ChrF$_B$	BLEU$_A$	BLEU$_B$
JDExploreAcademy	58.0	63.5	58.5	61.8	33.7	35.8
Online-B	56.9	63.6	58.3	61.9	33.3	36.6
Lan-Bridge	56.5	63.6	58.5	62.3	33.4	37.0
Online-G	55.2	61.7	58.7	62.5	33.7	36.5
Online-Y	54.6	61.4	58.0	61.9	32.9	36.3
Online-A	54.5	62.2	58.4	62.7	33.3	37.2
Online-W	54.3	61.7	57.7	61.7	32.6	36.0
PROMT	51.8	59.4	57.8	62.1	32.5	36.6
LT22	25.6	33.3	51.3	55.7	26.0	30.9

Table 39: Automatic metric scores for ja-en.

System	COMET$_A$ ↑	ChrF$_A$	BLEU$_A$
NT5	42.0	51.3	26.6
Online-W	41.2	51.7	27.8
JDExploreAcademy	40.6	50.1	25.6
Online-B	39.6	49.9	24.7
DLUT	37.2	49.8	24.8
NAIST-NICT-TIT	33.4	48.3	22.7
Online-A	32.9	48.4	22.8
LanguageX	32.9	49.1	22.4
Online-Y	32.3	48.2	21.5
Lan-Bridge	31.9	48.7	22.8
AISP-SJTU	30.1	48.0	22.0
Online-G	22.3	45.7	19.7
KYB	17.3	43.4	18.1
AIST	-152.7	11.4	0.1

Table 40: Automatic metric scores for liv-en.

System	COMET$_A$ ↑	ChrF$_A$	BLEU$_A$
TartuNLP	-5.8	53.5	29.9
TAL-SJTU	-8.4	53.2	30.4
HuaweiTSC	-27.3	48.4	23.4
Liv4ever	-44.0	46.7	23.3
NiuTrans	-88.3	35.6	13.0

Table 41: Automatic metric scores for ru-en.

System	COMET$_A$ ↑	ChrF$_A$	BLEU$_A$
Online-G	65.1	70.0	46.7
JDExploreAcademy	64.9	68.9	45.1
Online-Y	64.1	68.2	43.8
Lan-Bridge	63.1	68.5	45.2
Online-B	63.1	68.3	45.0
Online-A	62.2	68.3	43.9
Online-W	61.6	66.3	42.6
HuaweiTSC	60.9	68.5	45.1
SRPOL	59.5	67.2	43.6
ALMAnaCH-Inria	26.8	57.9	30.3

Table 42: Automatic metric scores for uk-en.

System	COMET$_A$ ↑	ChrF$_A$	BLEU$_A$
Online-B	62.5	67.2	44.4
Lan-Bridge	62.4	67.3	44.6
GTCOM	61.9	67.1	43.9
Online-G	57.4	66.0	43.2
Online-A	52.1	65.2	42.3
HuaweiTSC	50.1	63.9	41.6
Online-Y	49.8	64.6	41.8
PROMT	49.6	64.7	42.1
ARC-NKUA	49.6	64.6	41.9
ALMAnaCH-Inria	21.8	55.6	30.0

Table 43: Automatic metric scores for zh-en.

System	COMET$_A$ ↑	COMET$_B$	ChrF$_A$	ChrF$_B$	BLEU$_A$	BLEU$_B$
Online-G	45.6	36.2	59.7	54.1	29.6	21.7
JDExploreAcademy	45.1	35.2	61.1	54.1	33.5	22.3
LanguageX	44.9	35.3	60.5	54.2	31.9	22.1
Lan-Bridge	43.0	34.0	57.8	52.7	28.1	20.9
HuaweiTSC	42.8	33.5	58.5	52.8	29.8	21.7
Online-B	42.1	32.8	58.2	52.9	28.8	21.1
AISP-SJTU	41.6	32.8	59.2	53.8	29.7	21.4
Online-Y	40.8	31.0	57.6	52.1	27.1	19.8
Online-A	35.2	26.0	57.3	52.1	27.3	19.9
Online-W	31.6	23.1	54.5	49.9	24.0	18.0
NiuTrans	31.3	22.3	56.0	51.2	26.2	19.5
DLUT	30.6	22.0	55.2	50.5	25.0	18.6

Table 44: Automatic metric scores for cs-uk.

System	COMET$_A$ ↑	ChrF$_A$	BLEU$_A$
AMU	99.4	61.5	34.7
Online-B	94.3	64.0	38.3
GTCOM	93.4	63.9	36.8
Lan-Bridge	91.8	64.0	38.3
CharlesTranslator	90.8	61.5	34.3
HuaweiTSC	90.7	62.6	36.0
CUNI-JL-JH	90.0	61.6	34.8
Online-G	88.3	60.8	32.5
Online-A	87.8	62.2	35.9
CUNI-Transformer	87.3	61.6	35.0
Online-Y	78.4	59.6	32.1
ALMAnaCH-Inria	61.3	54.5	26.8

Table 45: Automatic metric scores for de-fr.

System	COMET$_A$ ↑	ChrF$_A$	BLEU$_A$
Online-B	70.5	74.6	58.4
Online-W	63.6	65.5	43.6
Online-Y	57.8	66.8	46.2
Online-A	52.2	64.5	41.3
Online-G	44.8	62.7	39.0
LT22	10.4	54.4	28.3

Table 46: Automatic metric scores for fr-de.

System	COMET$_A$ ↑	ChrF$_A$	BLEU$_A$
Online-W	77.9	81.2	64.8
Online-B	63.7	68.7	46.6
Online-Y	61.6	67.5	45.0
Online-A	59.2	67.2	44.4
eTranslation	55.4	68.4	46.5
Lan-Bridge	51.1	65.0	41.8
Online-G	48.2	66.0	41.1

Table 47: Automatic metric scores for ru-sah.

System	COMET$_A$ ↑	ChrF$_A$	BLEU$_A$
Online-G	-17.1	47.0	14.7
Lan-Bridge	-124.3	11.3	0.0

Table 48: Automatic metric scores for sah-ru.

System	COMET$_A$ ↑	ChrF$_A$	BLEU$_A$
Online-G	31.1	55.5	29.6
Lan-Bridge	-75.9	28.3	7.1

Table 49: Automatic metric scores for uk-cs.

System	COMET$_A$ ↑	ChrF$_A$	BLEU$_A$
AMU	104.8	60.7	37.0
Online-B	96.5	60.3	36.4
Lan-Bridge	94.5	60.4	36.5
HuaweiTSC	91.4	59.6	36.0
CharlesTranslator	90.2	59.0	35.9
CUNI-JL-JH	89.0	58.7	35.1
CUNI-Transformer	88.5	59.0	35.8
Online-A	85.4	57.5	33.3
Online-G	84.2	56.3	31.5
GTCOM	80.2	55.8	31.3
Online-Y	78.6	55.3	29.6
ALMAnaCH-Inria	62.4	50.7	25.3

Results of WMT22 Metrics Shared Task: Stop Using BLEU – Neural Metrics Are Better and More Robust

Markus Freitag(1), **Ricardo Rei**(2,3,4), **Nitika Mathur**(5), **Chi-kiu Lo**(6), **Craig Stewart**(2), **Eleftherios Avramidis**(8), **Tom Kocmi**(7), **George Foster**(1), **Alon Lavie**(2) **and André F. T. Martins**(2,3,9)
(1)Google Research (2)Unbabel (3)INESC-ID (4)Instituto Superior Técnico
(5)Oracle Digital Assistant (6)National Research Council Canada (7)Microsoft
(8)German Research Center for Artificial Intelligence (DFKI) (9)Instituto de Telecomunicações
wmt22-metric@googlegroups.com

Abstract

This paper presents the results of the WMT22 Metrics Shared Task. Participants submitting automatic MT evaluation metrics were asked to score the outputs of the translation systems competing in the WMT22 News Translation Task on four different domains: news, social, e-commerce, and chat. All metrics were evaluated on how well they correlate with human ratings at the system and segment level. Similar to last year, we acquired our own human ratings based on expert-based human evaluation via Multidimensional Quality Metrics (MQM). This setup had several advantages, among other things: (i) expert-based evaluation is more reliable, (ii) we extended the pool of translations by 5 additional translations based on MBR decoding or rescoring which are challenging for current metrics.

In addition, we initiated a challenge set subtask, where participants had to create contrastive test suites for evaluating metrics' ability to capture and penalise specific types of translation errors.

Finally, we present an extensive analysis on how well metrics perform on three language pairs: English→German, English→Russian and Chinese→English. The results demonstrate the superiority of neural-based learned metrics and demonstrate again that overlap metrics like Bleu, spBleu or chrF correlate poorly with human ratings. The results also reveal that neural-based metrics are significant better than non-neural metrics across different domains and challenges.

1 Introduction

The metrics shared task[1] has been a key component of WMT since 2008, serving as a way to validate the use of automatic MT evaluation metrics and drive the development of new metrics. We evaluate reference-based automatic metrics that score MT output by comparing the translations with a reference translation generated by human translators, who are instructed to translate "from scratch" without post-editing from MT. In addition, we also invited submissions of reference-free metrics (quality estimation metrics or QE metrics) that compare MT outputs directly with the source segments. All metrics are evaluated based on their agreement with human rating when scoring MT systems and human translations at the system or sentence level. The final ranking of this year's submitted primary metrics is shown in Table 1. We provide details in the remainder of the paper.

Metric	avg rank
MetricX XXL	1.20
Comet-22	1.32
UniTE	1.86
Bleurt-20	1.91
Comet-20	2.36
Matese	2.57
CometKiwi*	2.70
MS-Comet-22	2.84
UniTE-src*	3.03
YiSi-1	3.27
Comet-QE*	3.33
Matese-QE*	3.85
Mee4	3.87
BertScore	3.88
MS-Comet-qe-22*	4.06
chrF	4.70
f101spBleu	4.97
HWTSC-Teacher-Sim*	5.17
Bleu	5.31
Reuse*	6.69

Table 1: Official ranking of all primary submissions of the WMT22 Metric Task. The final score is the weighted average ranking over 201 different scenarios. Metrics with * are reference-free metrics.

We implemented several changes to the methodology that was followed in previous years' editions:

- **Expert-based human evaluation**: Like last year, we collected our own human ratings for select language pairs (en→de, en→ru, zh→en) from professional translators via MQM (Lommel et al.,

[1] https://wmt-metrics-task.github.io/

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 46 - 68
December 7-8, 2022 ©2022 Association for Computational Linguistics

2014). Freitag et al. (2021a) showed that expert-based MQM evaluations produce more reliable[2] scores when compared to the DA-based human ratings acquired by the WMT Translation task. This step was necessary as Freitag et al. (2021a) showed that the DA-based ground-truth is already of lower quality than some of our submissions (Section 3).

- **Additional Training Data**: We encouraged the participants to make use of existing MQM annotations for newstest2020 (Freitag et al., 2021a)[3], and the MQM annotations from the WMT21 Metrics Task (Freitag et al., 2021b) to improve and/or test their metrics.

- **Additional MT systems**: The primary use case for automatic metrics is guiding research to translations that are better than what we can generate right now. To address this scenario, we not only want to evaluate metrics on MT output that we are currently capable of generating, but also on translations that are better than the current WMT submissions. For that we need to add alternative translations that cover a wider space of possible translations. To address this, we added MT systems that were generated with MBR decoding or reranking (Section 2.2).

- **Challenge sets subtask**: In the main metrics task, the metrics are evaluated on MT systems translating test sets drawn from large sources of continuous text. In an effort to have a more fine-grained analysis on the strengths and weaknesses of the metrics, we introduced the concept of challenge sets. A challenge set consists of contrasting MT outputs, which have been deliberately devised or selected to include correct and incorrect translations of particular phenomena, along with their respective reference translation. The evaluation of every metric in this setup depends on its ability to rank the correct translations higher than their corresponding incorrect ones. Whereas a first version of challenge sets appeared in last year's metrics shared task (Freitag et al., 2021b), this year they appear for the first time as a subtask in a decentralized manner. Inspired by the *Build it or break it: The Language Edition* shared task (Ettinger et al., 2017), participants (the *Breakers*) had to submit their own test suites to test the robustness of MT metrics to particular phenomena that they choose. Our first edition of this subtask (Section 8) received four challenge set submissions covering a wide range of phenomena and languages.

- **Meta Evaluation**: A main aim of the metrics task is to rank the overall performance of various metrics. This requires some way of aggregating scores across different settings (language pair, domain, granularity etc.), in order to provide a balanced picture. Correlations with human scores have different ranges in different settings, so averaging them is not a good solution. Last year, we adopted a proposal by Kocmi et al. (2021) that involves taking the microaverage of a metric's accuracy in making pairwise system-ranking decisions across different settings. This is easy to interpret and reflects a common use-case for metrics, but because we have only three language pairs, and thus relatively few pairwise comparisons, it tends to place many metrics into large significance clusters (eg, 8 metrics in the top cluster last year, including CHRF but excluding COMET). In an effort to better discriminate, and to represent a broader set of use-cases, this year we computed the average rank of each metric across a large set of tasks (Section 5). This statistic has a clear interpretation, is justified by social choice theory (Colombo et al., 2022), and makes it easy to zoom into different subsets of tasks to provide finer-grained characterizations. To reflect the importance of the accuracy metric from last year, we define it as a single highly-important task (out of 201 tasks in total), with an overall weight of 25%.

- **MTME**: Similar to last year, all results in this paper are calculated with MTME[4]. We want to encourage every metric developer to use this tool to calculate scores for consistency and comparability going forward.

Our main findings are:

- Out of 13 reference-based metrics **BLEU is ranked last**, followed by F200SPBLEU and CHRF.

[2] DA is unreliable for high-quality MT output; ranks human translations lower than MT; correlates poorly with metrics. Expert-based MQM ranks human translations higher than MT and correlates generally much better with automatic metrics.

[3] https://github.com/google/wmt-mqm-human-evaluation

[4] https://github.com/google-research/mt-metrics-eval

- **Neural fine-tuned metrics are not only better, but also robust to different domains**. Furthermore, based on the results from the four submitted challenge sets, neural fine-tuned metrics exhibit superior performance when compared to lexical and embedding similarity metrics.
- Top performing metrics from previous years are still top-performers, being only outperformed by model ensembles or metrics based on considerably larger neural models.
- For the first time since 2008, there was no new purely lexical metric submission, which indicates that metric developers are moving away from lexical metrics.

The rest of the paper is organized as follows: Section 2 describes the additional MT systems. Section 3 presents an overview of the conducted expert-based human evaluation. Section 4 describes the metrics evaluated this year (baselines and participants). Section 5 describes the conducted meta-evaluation. Section 6 reports our main results. Section 7 summarizes our results for additional WMT22 Translation task language-pairs based on their Direct Assessment human evaluation. Section 8 presents a description of the submitted challenge sets along with their findings. Finally, Section 9 presents our most relevant conclusions.

2 Translation Systems

Similar to the previous years' editions, the source, reference texts, and MT system outputs for the metrics task are mainly derived from the WMT22 general MT Task. In addition to the MT system outputs from the WMT evaluation campaign, we added translations from six additional MT systems which we deemed interesting for evaluation.

2.1 WMT Test Sets

The general MT 2022 test set contains around 2000 segments for each translation direction. This year, the test sets cover 4 domains: news, social, conversational, and e-commerce. There are around 500 sentences for each domain resulting in reasonably balanced test sets. English sources are identical for both into-German and into-Chinese translation directions. The reference translations provided for the test sets are translated by professional translators. We have two reference translations for English→German and Chinese→English sponsored by Microsoft and one reference translation for English→Russian sponsored by Google. For more details regarding the news test sets, we refer the reader to the WMT22 General MT task findings paper (Kocmi et al., 2022a).

2.2 Additional MT Output

Similar to last year, we want to expand the pool of translations beyond the WMT submissions, which usually are quite similar to each other. We added translations based on M2M100 and translations generated with MBR decoding.

M2M100 1.2B As the field moves forward to large multilingual pre-trained models, we are interested in comparing such general-purpose large multilingual MT systems against direct submissions to the general MT task. Models such as MBART50 (Tang et al., 2021) and M2M100 (Fan et al., 2021) are publicly available, easy to use and have recently been used as baselines and/or as a backbone for new research. We tested both models on the newstest2021 and we decided to include M2M100 1.2B as an additional MT output as it yielded better automatic scores.

MBR Outputs Minimum Bayes Risk (MBR) decoding has recently gained attention in MT as a decision rule, with the potential to overcome some of the biases of MAP decoding in NMT (Eikema and Aziz, 2020; Müller and Sennrich, 2021; Eikema and Aziz, 2021; Freitag et al., 2022; Fernandes et al., 2022). MBR decoding centrally relies on a reference-based utility metric: its goal is to identify a hypothesis with a high estimated utility (expectation under model distribution) with the hope that a high estimated utility translates into a high actual utility (with respect to a human reference). MBR decoding is particularly interesting for reference-based metrics as it stress tests the metric, using it as a utility function.

This year, we added three different MBR runs using three different utility functions (BLEU, BLEURT-20, and COMET-20) as additional translations. Freitag et al. (2022) demonstrated that the translations generated with a neural-based utility (BLEURT-20, and COMET-20) generate translations that are not only better when compared to MAP decoding, but the resulting translations are also significantly different from both the beam search decoding and the MBR decoding output using BLEU as a utility function. To make it even more interesting for the metric task, for these MBR translation models we used a transformer-big baseline trained only on WMT22 bilingual training data. By not using the strongest NMT system, we hope to see interesting new errors in the translation output. To generate the candidate list for MBR decoding, we sampled 256 times from the model using unbiased ancestral sampling.

Reranking Outputs Complementary to MBR outputs, we were also interested in comparing and evaluating the quality produced by reranking approaches based on QE. Our hope is that QE based reranking would lead to translations that are lexically different than traditional beam search output and thus lead to more diverse translations for the same source sentences. For English→German and English→Russian we used the Fairseq WMT19 systems[5] (Ng et al., 2019) with Nucleus Sampling (Holtzman et al., 2019) to generate 200 candidate translations, from which we choose the best translation according to the Tune Reranker proposed in Fernandes et al. (2022). For Chinese→English we used the same process but replacing the NMT model with MBART50 (many-to-one) and using only 50 samples.

3 MQM Human Evaluation

Automatic metrics are usually evaluated by measuring correlations with human ratings. The quality of the underlying human ratings is critical and recent findings (Freitag et al., 2021a) have shown that crowd-sourced human ratings are not reliable for high quality MT output. Furthermore, an evaluation schema based on MQM (Lommel et al., 2014), which requires explicit error annotation, is preferable to an evaluation schema that only asks raters for a single scalar value per translation. Similar to last year, we decided to not use the human ratings from the WMT General MT task, and conducted our own MQM-based human evaluation on a subset of submissions and a subset of language pairs that are most interesting for evaluating current metrics. This not only had the advantage of more reliable ratings for a subset of language pairs, but also gave us the opportunity to add our own translations that might be challenging for current metrics and are not part of an WMT submission.

MQM is a general framework that provides a hierarchy of translation errors which can be tailored to specific applications. Google and Unbabel sponsored the human evaluation for this year's metrics task for a subset of language pairs using either professional translators (English→German, Chinese→English) or trusted and trained raters (English→Russian). The error annotation typology and guidelines used by Google's and Unbabel's annotators differ slightly and are described in the following two sections.

3.1 English→German and Chinese→English

Annotations for English→German and Chinese→English were sponsored and executed by Google, using 11 professional translators (7 for English→German, 4 for Chinese→English) having access to the full document context. Each segment gets annotated by a single rater. Instead of assigning a scalar value to each translation, annotators were instructed to label error spans within each segment in a document, paying particular attention to document context. Each error was highlighted in the text, and labeled with an error category and a severity. To temper the effect of long segments, we imposed a maximum of five errors per segment, instructing raters to choose the five most severe errors for segments containing more errors. Segments that are too badly garbled to permit reliable identification of individual errors are assigned a special *Non-translation* error. Error severities are assigned independent of category, and consist of *Major*, *Minor*, and *Neutral* levels, corresponding respectively to actual translation or grammatical errors, smaller imperfections and purely subjective opinions about the translation. Since we are ultimately interested in scoring segments, we adopt the weighting scheme shown in Table 2, in which segment-level scores can range from 0 (perfect) to 25 (worst). The final segment-level score is an average over scores from all annotators. For more details, exact annotator instructions and a list of error categories, we refer the reader to Freitag et al. (2021a) as the exact same setup was used for the WMT21 metrics task.

Severity	Category	Weight
Major	Non-translation all others	25 5
Minor	Fluency/Punctuation all others	0.1 1
Neutral	all	0

Table 2: Google's MQM error weighting.

[5] https://github.com/facebookresearch/fairseq/tree/main/examples/wmt19

3.2 English→Russian

The annotations for English→Russian were provided by Unbabel who utilized four professional, native language annotators with ample translation experience. Annotation was conducted using Unbabel's own proprietary variant of the MQM framework (Lommel et al., 2014) which is fully compliant with MQM 2.0, being the most recent iteration of the framework[6]. Annotation was split along the four domain boundaries with each of the annotators evaluating all of the systems for a single content type. Similarly to Google, the annotators were given the full document context (up to ten segments) and were instructed to identify (by highlighting) and classify errors in accordance with the MQM typology. Annotators were also asked to classify error severity; in addition to *Minor* and *Major* error severities used by Google, Unbabel also uses a *Critical* error severity. However, in the interest of maintaining consistency in evaluation, we calculated the MQM score in a manner compliant with the Google methodology outlined above. Specifically all annotated *Critical* errors were counted as *Major* and punctuation errors were weighted using the weighting scheme in Table 2.

3.3 Human Evaluation Results

As discussed in Section 1, we decided to run our own human evaluation in order to generate our golden-truth ratings and come to stronger conclusions about the quality of each automatic metric across all domains. However, this also meant that we were only able to evaluate a subset of the test sets. In Table 3, you can see the number of segments for each language pair and test set that we used for human evaluation. We followed a simple and consistent approach to downsample the data: we kept the first 10 sentences of each document. By doing this, we did not need to discard any documents and only needed to crop longer documents. An exception is Chinese→English where we evaluated the full test set.

language	news	social	ecomm.	conv.
en→de	300/511	340/512	230/530	445/484
en→ru	300/511	340/512	230/530	445/484
zh→en	505/505	503/503	518/518	349/349

Table 3: Numbers of MQM-annotated segments per domain.

The results of the MQM human evaluation can be seen in Table 4. Most of the reference translations are ranked first, except for refB for English→German. Not ranking the human evaluation on top of the MT output is usually a signal for a corrupt human evaluation. We double checked the annotation for refB and can confirm that the reference translation indeed contained some errors.

4 Baselines and Primary Submissions

We computed scores for several baseline metrics in order to compare submissions against previous well-studied metrics. We will start by describing those baselines and then we will describe the submissions from participating teams. An overview of the evaluated metrics can be seen in Table 5.

4.1 Baselines

SacreBLEU baselines We use the following metrics from the SacreBLEU (Post, 2018) as baselines:

- **BLEU (Papineni et al., 2002)** is based on the precision of n-grams between the MT output and its reference weighted by a brevity penalty. Using SacreBLEU we obtained sentence-BLEU values using the `sentence_bleu` Python function and for corpus-level BLEU we used `corpus_bleu` (both with default arguments[7]).

- **F101SPBLEU (Goyal et al., 2022)** and **F200SPBLEU (NLLB Team et al., 2022)** are BLEU scores computed with subword tokenization done by standardized Sentencepiece Models (Kudo and Richardson, 2018). We used the command line SacreBLEU to compute the sentence level F101SPBLEU[8] and F200SPBLEU[9] and we average those scores to obtain a corpus-level score.

- **CHRF (Popović, 2015)** uses character n-grams instead of word n-grams to compare the MT output with the reference. For CHRF we used the SacreBLEU `sentence_chrf` function (with default arguments[10]) for segment-level scores and we average those scores to obtain a corpus-level score.

[6] https://themqm.org/

[7] nrefs.1|case.mixed|lang.LANGPAIR|tok.13a|smooth.exp |version.1.5.0

[8] nrefs:1|case:mixed|eff:yes|tok:flores101|smooth:exp| version:2.3.1

[9] nrefs:1|case:mixed|eff:yes|tok:flores200|smooth:exp| version:2.3.1

[10] chrF2|lang.LANGPAIR|nchars.6|space.false|version.1.5.0

	English→German ↓				
System	all	news	social	ecom.	conv.
refA	0.64	0.97	0.68	0.56	0.42
Online-W	0.79	0.95	0.74	0.93	0.65
refB	0.91	1.38	0.93	1.17	0.46
MBR-bleu	0.96	1.29	1.14	0.82	0.67
Online-B	1.04	1.44	1.27	0.88	0.67
JDExploreAcademy	1.05	1.36	1.21	1.20	0.64
MBR-comet	1.08	1.40	1.33	1.01	0.71
MBR-bleurt	1.11	1.55	1.41	0.72	0.78
Online-A	1.21	1.40	1.55	1.35	0.76
Online-G	1.22	1.78	1.51	1.17	0.66
Online-Y	1.30	1.99	1.45	1.02	0.86
QUARTZ	1.34	1.85	1.59	1.10	0.94
Lan-Bridge	1.41	2.43	1.72	1.09	0.65
OpenNMT	1.68	1.98	2.14	1.73	1.09
PROMT	1.76	2.41	1.94	1.56	1.27
M2M100	2.82	3.46	2.99	2.94	2.19

	Chinese→English ↓				
System	all	news	social	ecom.	conv.
refA	1.22	1.42	1.10	1.42	0.82
refB	2.00	2.18	1.83	1.69	0.96
Lan-Bridge	2.47	2.45	1.97	3.55	1.39
MBR-bleurt	2.51	2.52	2.06	3.68	1.55
Online-B	2.71	2.66	2.07	3.73	1.55
LanguageX	2.74	2.74	2.46	3.78	1.58
JDExploreAcademy	2.83	2.84	2.56	3.81	1.60
MBR-comet	2.87	2.88	2.63	3.98	1.61
Online-G	2.93	2.90	2.73	4.16	1.63
MBR-bleu	3.00	2.94	2.77	4.22	1.64
HuaweiTSC	3.09	2.96	2.80	4.30	1.68
AISP-SJTU	3.19	3.08	2.89	5.03	1.76
Online-Y	3.28	3.27	3.03	5.20	1.79
Online-A	3.73	3.49	3.48	5.39	2.04
Online-W	3.95	3.96	3.60	5.76	2.30
M2M100	6.82	7.47	5.78	9.37	3.61

	English→Russian ↓				
System	all	news	social	ecom.	conv.
refA	1.13	0.43	2.17	1.95	0.39
Online-W	1.37	1.35	2.96	0.90	0.41
MBR-bleu	1.85	1.57	4.01	1.39	0.63
Online-B	1.94	1.59	4.29	1.37	0.68
Online-G	2.03	1.50	4.33	1.88	0.71
JDExploreAcademy	2.09	1.14	4.63	2.23	0.71
MBR-comet	2.10	2.01	4.74	1.26	0.57
Lan-Bridge	2.34	2.14	5.49	1.49	0.51
Online-Y	2.55	2.06	5.79	1.66	0.86
Online-A	2.85	1.83	6.56	2.62	0.83
PROMT	2.94	2.04	6.88	2.55	0.73
HuaweiTSC	3.40	1.72	8.07	3.02	1.17
SRPOL	3.68	2.02	8.19	3.53	1.43
eTranslation	3.79	2.30	8.54	3.49	1.32
QUARTZ	4.06	3.82	7.02	5.03	1.46
M2M100	4.56	3.74	9.27	4.42	1.58

Table 4: MQM human evaluations for generaltest2022. Lower average error counts represent higher MT quality.

BERTScore (Zhang et al., 2020) leverages contextual embeddings from pre-trained transformers to create soft-alignments between words in candidate and reference sentences using cosine similarity. Based on the alignment matrix, BERTScore returns a precision, recall and F1 score. We used F1 without TF-IDF weighting.

YiSi-1 (Lo, 2019) is a MT evaluation metric that measures the semantic similarity between a machine translation and human references by aggregating the IDF-weighted lexical semantic similarities based on the contextual embeddings extracted from pre-trained language models (e.g. RoBERTa, CamemBERT, XLM-RoBERTa, etc.).

Bleurt (Sellam et al., 2020) is a learned metric that is fine-tuned to produce a DA for a given translation by encoding it jointly with its reference. We used the Bleurt20 checkpoint (Pu et al., 2021) which was trained on top of RemBERT using DA from previous shared tasks ranging 2015 to 2019 and additional synthetic data created from Wikipedia articles.

Comet (Rei et al., 2020) is a learnt metric that is fine-tuned to produce a z-standardized DA for a given translation by comparing its representation to source and reference embeddings. We used the default model `wmt20-comet-da` provided in version 1.1.2 which is trained on top of XLM-R large using data from from previous shared tasks ranging 2017 to 2019.

Comet-QE (Rei et al., 2021) is a reference-free learnt metric similar to Comet. We used the `wmt21-comet-qe-mqm`) model which was a top-performing metric from last year's shared task. This metric is first trained on z-standardized DA from 2017 to 2020 and then fine-tuned on z-standardized MQM from (Freitag et al., 2021a).

	metric	broad category	sup-erv.	ref. free	citation	availability (https://github.com/)
baselines	BLEU	lexical overlap			Papineni et al. (2002)	mjpost/sacrebleu
	F101spBLEU	lexical overlap			Goyal et al. (2022)	mjpost/sacrebleu
	F200spBLEU	lexical overlap			NLLB Team et al. (2022)	mjpost/sacrebleu
	chrF	lexical overlap			Popović (2015)	mjpost/sacrebleu
	BERTScore	embedding similarity			Zhang et al. (2020)	Tiiiger/bert_score
	BLEURT	fine-tuned metric	✓		Sellam et al. (2020)	google-research/bleurt
	Comet	fine-tuned metric	✓		Rei et al. (2020)	Unbabel/COMET
	Comet-QE	fine-tuned metric	✓	✓	Rei et al. (2021)	Unbabel/COMET
	YiSi-1	embedding similarity			Lo (2019)	chikiulo/yisi
primary submissions	Comet-22	fine-tuned metric	✓		Rei et al. (2022)	Unbabel/COMET
	CometKiwi	fine-tuned metric	✓	✓	Rei et al. (2022)	Unbabel/COMET
	EE-BERTScore	embedding similarity			Liu et al. (2022)	(not available)
	KG-BERTScore	embedding similarity		✓	Liu et al. (2022)	(not available)
	MaTESe	fine-tuned metric	✓		Perrella et al. (2022)	(not available)
	MaTESe-QE	fine-tuned metric	✓	✓	Perrella et al. (2022)	(not available)
	MEE4	lexical & embedding similarity			Mukherjee and Shrivastava (2022b)	AnanyaCoder/WMT22Submission
	MetricX XXL	fine-tuned metric	✓			(not available)
	MS-Comet	fine-tuned metric	✓		Kocmi et al. (2022b)	MicrosoftTranslator/MS-Comet
	MS-Comet-QE	fine-tuned metric	✓	✓	Kocmi et al. (2022b)	MicrosoftTranslator/MS-Comet
	REUSE	embedding similarity		✓	Mukherjee and Shrivastava (2022a)	AnanyaCoder/WMT22Submission_REUSE
	Teacher-Sim	fine-tuned metric	✓	✓	Liu et al. (2022)	(not available)
	SEScore	fine-tuned metric			Xu et al. (2022)	xu1998hz/SEScore
	UniTE	fine-tuned metric	✓		Wan et al. (2022b)	NLP2CT/UniTE

Table 5: Baseline metrics and primary submissions for the metrics task. We categorize metrics into 3 major classes: lexical, embedding similarity and fine-tuned metrics. Regarding fine-tuned metrics we have metrics that use human quality scores such as DA or MQM and metrics that use synthetic labels for fine-tuning (3rd column).

4.2 Metric Submissions

The rest of this section summarizes participating metrics. The ★ symbol indicates that the metric is the primary submission of the research group.

Comet-22★ (Rei et al., 2022) is an ensemble of two models; 1) Comet estimator model trained with Direct Assessments and 2) a newly proposed multitask model trained to predict sentence-level MQM scores along with Ok/Bad word-level tags derived from annotation spans.

CometKiwi★ ensembles 2 QE models similarly to COMET-22; 1) classic Predictor-Estimator QE model trained on DAs ranging 2017 to 2019 and then fine-tuned on DAs from MLQE-PE (the official DA from the QE shared task) and 2) the same multitask model used in the COMET-22 submission but without access to a reference translation.

MS-Comet-22★ and MS-Comet-QE-22★ (Kocmi et al., 2022b) are built on top of Comet by Microsoft Research using proprietary data. This metric is trained on a several times larger set of human judgements compared to Comet-baseline, covering 113 languages and 15 domains. Furthermore, the authors propose filtering of human judgement with potentially low quality to further improve the model.

MS-Comet-22 evaluated source, MT hypothesis and human reference from the input, while MS-Comet-QE-22 calculated scores in quality estimation fashion with only source segment and MT hypothesis.

EE-BERTScore★ (Liu et al., 2022) stands for Entropy Enhanced BERTScore and aims at achieving a more balanced system-level rating by assigning weights to segment-level scores produced by BERTScore. The weights are determined by the difficulty of a segment determined by the entropy between the hypothesis-reference pair.

KG-BERTScore (Liu et al., 2022) is a reference-free machine translation (MT) evaluation metric, which incorporates multilingual knowledge graph into BERTScore by linearly combining the results of BERTScore and bilingual named entity matching.

Cross-QE (Liu et al., 2022) is a reference-free metric with a similar architecture to Comet-QE.

HWTSC-Teacher-Sim★ (Liu et al., 2022) is a reference-free metric by fine-tuning the multilingual Sentence BERT model paraphrase-multilingual-mpnet-base-v2

HWTSC-TLM (Liu et al., 2022) is a reference-free metric which only uses a target-side language model to score the system translations as input.

Matese★ (Perrella et al., 2022) and Matese-QE★ leverage transformer-based multilingual encoders to identify error spans in translations, and classify their severity between *Minor* and *Major*. The quality score returned for a translation is computed following the MQM error weighting used by Google (see Section 3.1).

Mee (Mukherjee et al., 2020) is an automatic evaluation metric that leverages the similarity between embeddings of words in candidate translation and the corresponding reference. Unigrams are matched based on their surface forms, root forms and meanings while semantic evaluation is achieved by using pretrained fasttext embeddings. MEE computes evaluation score using three modules namely exact match, root match and synonym match. In each module, fmean-score is calculated giving more weight to recall. Final score is the average of the three individual modules.

Mee2 and Mee4★ (Mukherjee and Shrivastava, 2022b) are improved versions of Mee focusing on computing contextual and syntactic equivalences along with lexical, morphological and semantic similarity. The intent is to capture fluency and context of the MT outputs along with their adequacy. Fluency is captured using syntactic similarity and context is captured using sentence similarity leveraging sentence embeddings. The final score is the weighted combination of three similarity scores: a) syntactic similarity achieved by modified Bleu score; b) lexical, morphological and semantic similarity: measured by explicit unigram matching; c) contextual similarity: sentence similarity scores from Language-Agnostic BERT model.

Reuse★ (Mukherjee and Shrivastava, 2022a) is a bilingual, unsupervised reference-free metric. It estimates the translation quality at chunk-level and sentence-level. Source and target sentence chunks are retrieved by using a multi-lingual chunker. Chunk-level similarity is computed by leveraging BERT contextual word embeddings and sentence similarity scores are calculated by leveraging sentence embeddings of Language-Agnostic BERT models. The final quality estimation score is obtained by mean pooling the chunk-level and sentence-level similarity scores.

MetricX XL and MetricX XXL★ are massive multi-task metrics, which fine-tune large language model checkpoints such as mT5 on a variety of human feedback data such as DA, MQM, QE, NLI and Summarization Eval. The resulting primary submission uses the MQM score outputted by a fine-tuned 30B mT5.

UniTE★ (Wan et al., 2022a,b) is a learnt metric that can possess the ability of evaluating translation outputs following all three evaluation scenarios, i.e., source-only, reference-only, and source-reference-combined. Following their previous work, the authors improve their models by pre-training on pseudo-labeled data examples, and applying data cropping and a ranking-based score normalization during fine-tuning. The resulting submission is an ensemble of two models trained with different backbone models (XLM-R and InfoXLM).

SEScore★ (Xu et al., 2022) is an unsupervised reference-based evaluation metric, which takes model output and reference to produce a quality score. SEScore is trained from a pre-trained language model (Ex. Roberta) on synthetic triples generated from raw text. The synthetic triples consist of (raw text, synthetic error text, pseudo score), corresponding to (reference, model output, human rating). The data used for training the metric is constructed by synthesising candidate sentences y' to mimic plausible errors by transforming raw input sentences multiple times. At each step, a random span of text is selected and new content is inserted, deleted or replaced. All these errors are non-overlapping. The authors name this data construction process "stratified error synthesis", which randomly samples a set of potential errors and stochastically applies them on a given sentence. The score assigned to the perturbed sentences is a raw count of the severities applied by each transformation. In the end, SEScore is a regression quality prediction model trained on synthetic triples. Since this process can be applied to raw data and the resulting model can be developed for any text generation domain.

5 Meta Evaluation

Our main goal in evaluating metrics is to establish a ranking that reflects a metric's accuracy across a broad range of settings and applications. Combining results across different settings is challenging because correlations with human gold scores have different ranges and may be subject to differing degrees of noise. There are also many ways of measuring correlation, with different strengths and weaknesses, and it is often not clear which is best in a given setting.

This year, our overall ranking is just each metric's average rank across a large number of "tasks". Unlike raw correlation scores, ranks are comparable across tasks. The resulting global ranking approximates the "Kemeny consensus" – the ranking with lowest aggregate Kendall distance to the per-task rankings – which in turn satisfies several criteria from social choice theory (Colombo et al., 2022). Our version has the following features:

- We use a large number of tasks which may contain overlapping information. For instance, on each dataset, we compute both Pearson and Kendall-Tau correlation, and treat these as separate tasks. This makes the overall ranking robust to quirks in particular correlations.
- To guard against inadvertent bias toward settings that have more tasks than others, we use a task weighting that reflects the relative importance of various attributes (language pair, domain, etc.).
- Within each task, we establish a ranking that includes ties to reflect statistical significance. This naturally up-weights tasks that are more discriminative. For instance, a task that yields the ranking 1, 1, 1, 1 will not affect the overall ranking at all, while a ranking of 1, 2, 3, 4 is a maximal vote.
- In order to indicate metric proximity, we report raw averages over (weighted) per-task ranks rather than the resulting ranking as advocated by Colombo et al. (2022). For instance, average ranks of 1.1, 1.2, 2.1, 3.9 indicate that the top two metrics perform similarly and the last metric is considerably worse; these details is lost in the global ranking 1, 2, 3, 4.
- We also report rankings on selected subsets of tasks to characterize metric behavior on attributes such as language or domain.

5.1 Tasks

Tasks are identified by unique value assignments for each of the following attributes: language, domain, level, include-human, averaging method, and correlation. These are as follows:

Language (4 values)

Language pairs include those for which we have MQM ratings – English→German, English→Russian, and Chinese→English – plus *All*, which indicates all pairs pooled together.

Domain (5 values)

We computed correlations on domain-specific portions of each test-set as well as on each test-set as a whole. All language pairs have the same set of domains: *conversation*, *e-commerce*, *news*, and *social*. We use *mixed* to refer to all domains together, *i.e.*, the whole test set.

Level (2 values)

For each domain (including *mixed*), we computed correlations at the *system* level and the *segment* level. Human scores for each domain are averages over the corresponding segments. For metric submissions that did not include domain-level scores, we computed similar averages.

Include-human (2 values)

We computed separate correlations over sets of outputs that exclude human references (include-human=*false*) and that include all available references (include-human=*true*) except the standard reference, which is never scored by metrics. The first scenario reflects the standard use-case for metrics; the second captures a future scenario in which MT output quality approaches human quality. Since English→Russian has only a single reference, it participates only in the first condition. For the other two language pairs we use the reference that was judged best by the MQM raters. Table 6 summarizes the use of reference translations for different language pairs.

language	best ref	scored ref
en→de	A	B
en→ru	A	{}
zh→en	A	B

Table 6: Use of reference translations.

language	domain	level	+human	averaging	correlation	tasks	weight
all (1/4)	mixed (1/1)	sys (1/1)	no (1/1)	none (1/1)	acc (1/1)	1	1/4
en-ru (1/4)	* (1/5)	sys (1/2)	no (1/1)	none (1/1)	P,K (1/2)	10	1/80
		seg (1/2)	no (1/1)	* (1/3)	P,K (1/2)	30	1/240
en-de,zh-en (1/4)	* (1/5)	sys (1/2)	* (1/2)	none (1/1)	P,K (1/2)	40	1/160
		seg (1/2)	* (1/2)	* (1/3)	P,K (1/2)	120	1/480
						201	

Table 7: Task weighting. Column entries are sets of values for the attribute in the heading, with * designating all possible values. Numbers in brackets show the weight assigned to each value in the set. Each line corresponds to a set of tasks that have the same weight: the product of all the per-attribute weights shown in brackets. *P* and *K* refer to Pearson and Kendall correlation, respectively.

Averaging (3 values)

At the segment level, metric and human scores are naturally represented as system × segment matrices. However, correlations operate over pairs of vectors rather than pairs of matrices. There are three ways to resolve the problem: flatten the matrices into single vectors, compute average correlations over matching pairs of row vectors, or compute average correlations over matching pairs of column vectors. We designate these as *none*, *system*, and *segment* averaging, respectively. They measure a metric's ability to rate an arbitrarily-chosen (system, segment) pair, an arbitrary segment for a fixed system, and different system outputs for the same segment. Last year we used only the first alternative; this year include all three. System-level correlations do not require averaging, since their inputs are vectors in the first place.

Correlation (3 values)

We computed three correlations: system-level pairwise ranking *accuracy* (as proposed by Kocmi et al., 2021), *Pearson* and *Kendall*. Accuracy was used only for a single task in which all language pairs were pooled (language=*All*), while Pearson and Kendall were used for all other tasks. Pearson correlation tests linear fit with MQM scores, a stringent but reasonable criterion since we expect these scores to conform to a linear scale (for example, a translation with two minor errors is twice as bad as one with only a single error). Pearson has well-known drawbacks (Mathur et al., 2020), notably sensitivity to outliers, which we minimized by choosing only relatively high-performing systems. Like accuracy, Kendall is based on pairwise score comparisons, and thus reflects a common ranking use-case. It is susceptible to noise in gold pairwise rankings, for which a common strategy is to discard pairs judged not to be significantly different. We did not take this into account, relying instead on our significance tests for metric (rather than system) rankings.

5.2 Task Weighting

As explained in the previous section, attributes are not independent. For instance, there are three averaging methods for segment-level tasks, but only one for system-level tasks. If all tasks were weighted equally, this would have the undesirable consequence of making segment-level correlations count for 3× as much as system-level correlations when determining the overall ranking.

To avoid this, we used a hierarchical weighting scheme. We first ordered the attributes as listed in the previous section, then distributed weights evenly among all permissible values at each step of the hierarchy. The results are shown in Table 7. There are a total of 201 tasks, of which the accuracy task for all language pairs receives a weight of 1/4, with the remaining mass of 3/4 distributed among tasks whose individual weights vary between 1/80 and 1/420.

In Figures 1 through 4, we show analyses of how metric performance varies along different dimensions (attributes) such as language, domain, etc.. To do this, we partition tasks according to the values of the selected attribute, re-normalizing their global weights so they sum to 1 for each partition. We then compute weighted average ranks for each partition separately, in the same fashion as the overall ranking.

5.3 Per-task Ranking

For each task, we compare all pairs of metrics, and determine whether the difference in their correlation scores is significant according to the PERM-

BOTH hypothesis test of Deutsch et al. (2021), using 1000 re-sampling runs, and setting $p = 0.05$. For the averaging methods, sampling is performed separately for each row or column vector prior to averaging.

We then assign ranks as follows. Starting with the highest-scoring metric, we move down the list of metrics in descending order by score, and assign rank 1 to all metrics until we encounter the first metric that is significantly different from any that have been visited so far. That metric is assigned rank 2, and the process is repeated. This continues until all metrics have been assigned a rank.

6 Main Results

As we have seen in Section 5, the main results are defined across different settings including system-level and segment-level tasks. Nonetheless, since the main use case of automatic metrics is to rank systems, system-level accuracy has a 1/4 weight on the final score with the remaining 3/4 distributed over 200 different settings.

Table 1 shows the official ranking of all primary submissions over the 201 different settings. A key observation is that neural metrics perform significantly better than lexical metrics. Of the 20 evaluated metrics, BLEU and SPBLEU are ranked 19th and 17th respectively. On the other hand, fine-tuned neural baseline metrics such as COMET-20 and BLEURT-20 are still ranked above several of the new primary submissions. They are outperformed only by submissions based on models that are considerably larger[11]. Figure 1 shows the ranking split by the different language pairs. The trend is very similar for all language pairs. While METRICX XXL performs best for En→De and En→Ru, COMET-22 performs best for Zh→En.

One open question about neural metrics has been their ability to generalise to new domains, since most training and testing data from previous years were based on News data. In Figure 2 we present the performance of each metric across four domains: news, social, conversational, and e-commerce. Similar to last year, we observe that the neural metrics perform better than lexical overlap metrics across all four domains.

Figure 3 shows the average rankings when grouped separately by system-level and segment-level tasks. Many metrics fall into the same significance cluster when evaluated on the system-level as we only have a very limited number of MT systems. Nevertheless, we observe that the metric rankings are largely stable across both granularities and that METRICX XXL and COMET-22 perform best on both the segment-level and system-level tasks. The differences are more prevalent in the segment-level task, though.

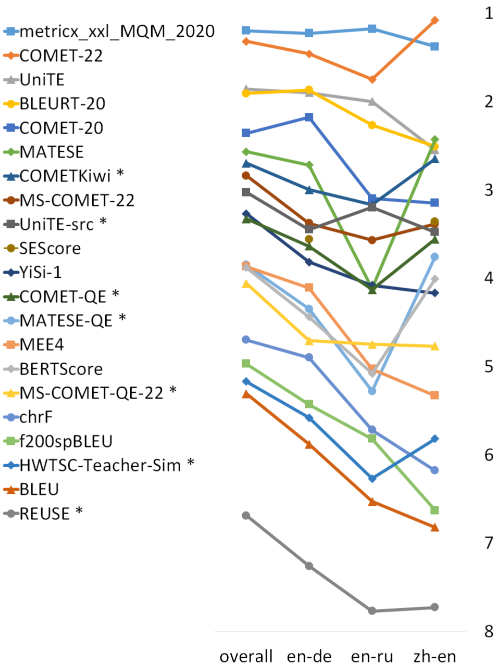


Figure 1: Weighted ranking of metrics' correlation with human grouped by translation directions.

In Figure 4, we compare the rankings when including human translations as MT systems (with human) or just considering MT submission (without human). Overall, the majority of metrics show

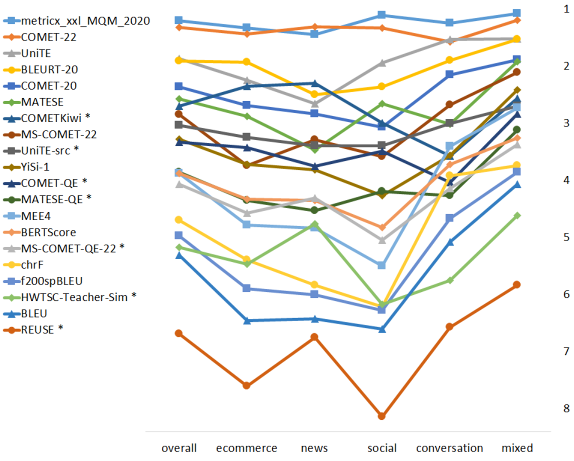


Figure 2: Weighted ranking of metrics' correlation with human grouped by domains.

[11] Both UNITE and COMET-22 are ensembles of two models trained on XLM-R variants while METRICX XXL uses mT5 XXL as a backbone

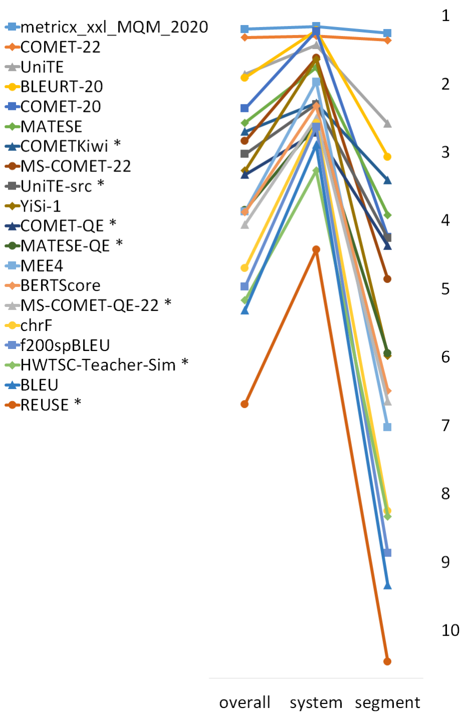


Figure 3: Weighted ranking of metrics' correlation with human grouped by granularity levels.

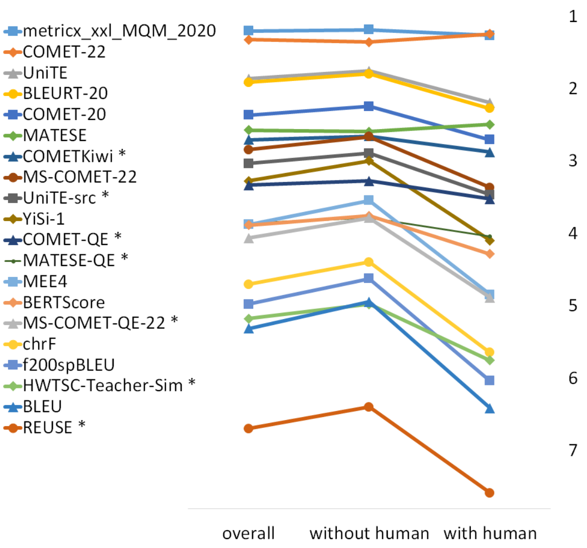


Figure 4: Weighted ranking of metrics' correlation with human grouped by candidate pools (with or without human translations).

lower correlation when we include human translations, except Comet-22 and Matese.

7 Direct Assessment Human Evaluation

In addition to our MQM annotations and as a contrastive evaluation to cover more language pairs, we look into the performance of metrics when compared to the human evaluation campaign conducted by the General MT shared task (Kocmi et al., 2022a), who ran human evaluation for all 21 translation directions and WMT22 submissions. Last year, we decided to exclude the human ratings by the WMT main task as they were of lower quality than the best automatic metrics. However, the GeneralMT task improved their evaluation methodology in particular for all from-English and non-English translation directions and implemented the Scalar Quality Metric (SQM) which has been shown to have high correlation with MQM on at least the system-level (Freitag et al., 2021a). The GeneralMT task used two different human evaluation methodologies depending on the language pair: reference-based Direct Assessment (Ref. DA) (Graham et al., 2013) and SQM style source-based DA (DA+SQM) (Kocmi et al., 2022a).

Ref. DA has been used for all into-English translation directions and asks human raters to judge each system translation against human reference translation on a 0–100 scale. This technique does not use bilingual speakers and is evaluated by non-professional crowd workers. In order to increase quality of assessment, there are several quality control items. Out of all collected human annotations, 63% have been removed due to failing quality control.

DA+SQM asks bilingual raters to annotate system translations against original sources on a 0–100 labeled scale. The scale is marked with seven points representing expected quality. In this setting, Kocmi et al. (2022a) evaluated all from-English and non-English translation directions. They used mainly professional raters.

We present system-level accuracy results in Table 8. The ranking generated based on accuracy scores when taking the DA+SQM annotation as ground truths is comparable to the primary results in Table 1, ranking MetricX XXL as the best performing metric followed by UniTe and COMET-22. Similarly, it ranks n-gram matching metrics (Bleu, chrF, f101spBleu) among worst performing metrics. This confirms the main findings from MQM evaluation.

On the other hand, accuracy scores taking ref. DA as the ground truth, result in a very different ranking of the metrics. It ranks n-gram matching metrics as the top performing metrics. This suggest that the technique does not evaluate systems well

Number of languages	13	6
Number of system pairs	564	329
Human judgement style	DA+SQM	ref. DA
MetricX XXL	**0.862** (1)	0.620 (11)
UniTE	0.849 (2)	0.623 (10)
Comet-22	0.842 (3)	0.626 (9)
CometKiwi*	0.835 (4)	0.617 (12)
MS-Comet-22	0.833 (5)	0.626 (9)
Bleurt-20	0.830 (6)	0.650 (5)
Comet-20	0.826 (7)	0.635 (8)
MS-Comet-qe-22*	0.824 (8)	0.641 (7)
Comet-QE*	0.821 (9)	0.605 (13)
UniTE-src*	0.800 (10)	0.623 (10)
YiSi-1	0.785 (11)	0.660 (3)
BERTScore	0.764 (12)	0.666 (2)
chrF	0.762 (13)	0.666 (2)
EE_BERTScore	0.750 (14)	0.647 (6)
f101spBleu	0.748 (15)	**0.669 (1)**
HWTSC-Teacher-Sim*	0.720 (16)	0.568 (15)
Bleu	0.707 (17)	0.653 (4)
Reuse*	0.344 (18)	0.584 (14)

Table 8: System-level pairwise accuracy for WMT style human evaluation. Numbers in brackets show rank of metrics given human judgement style. The highest score is present bolded.

and instead human crowd workers are incentivized to quickly compare the surface forms of translation against reference without understanding. We would advise metric developers and researchers running human evaluations not to use reference-based DA, especially when evaluated with non-professional crowd workers.

8 Challenge Sets Subtask

The challenge sets subtask is inspired by the *Build it or break it: The Language Edition* shared task (Ettinger et al., 2017) which aimed at testing the generalizability of NLP systems beyond the distributions of their training data. With that said, our goal is to encourage researchers to build a set of test sets that measure metrics' ability to detect different targeted phenomena that might not be well represented in traditional test sets used to evaluate metrics.

This subtask is made of three consecutive phases; 1) the *Breaking Round*, 2) the *Scoring Round* and 3) the *Analysis Round*:

1. In the *Breaking Round*, the challenge set participants (*Breakers*) submit their challenge sets composed of contrastive examples for different phenomena with source sentences (s), incorrect translations ($\hat{t}$), correct translations (t) and references (r).

2. In the *Scoring Round* the metrics participants from the main task (the *Builders*) are asked to score all translations with their metrics without knowing which ones are correct or incorrect. Also, in this phase the organisers score all data with the baseline metrics.

3. Finally, after gathering all metric scores, the data is returned to the *Breakers* for the *Analysis round*, where they look at which metrics are able to correctly rank the correct translations above the incorrect ones for the different phenomena being tested.

We had a total of 4 submissions to this shared task, covering a wide range of phenomena and 146 different language pairs. Table 9 provides an overview of the submitted challenge sets. A short description of every submission follows:

ACES The ACES (Translation Accuracy Challenge Sets; Amrhein et al., 2022) results from a collaboration between the University of Zurich with the University of Edinburgh. This challenge set, highly inspired by the MQM framework, consists of 36,499 examples, covering 146 language pairs and 68 phenomena, ranging from simple perturbations at the word/character level to more complex errors based on discourse and real-world knowledge. The data was created artificially for some error types and manually for others.

Their analysis aimed to reveal the extent to which metrics take into account the source sentence context and the surface-level overlap with the reference, and if they profit by using multilingual embeddings. Finally, they recommend that one considers a) **combining metrics with different strengths** and b) explicitly **modelling additional language-specific information** beyond what is available via multilingual embeddings.

SMAUG The challenge set based on Sentence-level Multilingual data Augmentation (SMAUG; Alves et al., 2022), submitted by Unbabel and IST evaluates the robustness of MT metrics to 5 different types of translation errors; Named entity errors, numerical errors, meaning errors, insertion of content and content missing. These errors are created by perturbing reference translations and then curated by the authors. The challenge set covers 3

challenge set	method	lang. pairs	pheno-mena	items	citation	availability (https://github.com/)
ACES	automatic	146	68	36,499	Amrhein et al. (2022)	EdinburghNLP/ACES
DFKI-CS	semi-autom.	2	107	19,347	Avramidis and Macketanz (2022)	DFKI-NLP/mt-testsuite
HwTsc-CS	semi-autom.	1	5	721	Chen et al. (2022)	HwTsc/Challenge-Set-for-MT-Metrics
SMAUG	automatic	3	5	632	Alves et al. (2022)	Unbabel/smaug

Table 9: Overview of the participations at the challenge sets task

language pairs and contains close to 50 high-quality examples for each phenomenon.

In this challenge set the authors show that there has been a promising progress in terms of detecting these critical errors when compared to last year's metric submissions. Nevertheless, errors related to **named entities and numbers were found to pose a challenge for several tested metrics**. Also, due to a high variance in the observed results across all the error types it becomes **hard to predict performance of current methods with respect to untested translation errors**.

HWTSC Challenge Set The challenge set submitted by Huawei Translation Services Center (Chen et al., 2022) aims at examining metrics ability to handle synonyms and to discern critical errors in translations. This challenge set is composed of 721 zh-en examples for 5 different error types; Named entity errors, numerical errors, time & date errors, wrong unit conversions and Affirmation/Negation errors. The underlying data is either WMT 21 or Flores 101 which covers two distinct domains, News and Wikipedia respectively. To create alternative translations the authors used in-house translators (performing post-edit) and to create the adversarial translations they used LIST (Alzantot et al., 2018).

The authors of this challenge set conclude that although embedding-based metrics perform relatively well on discerning sentence-level negation/affirmation errors, they **perform poorly on relating synonyms**. Additionally they find that the generalizability of some metrics is compromised, as they are **susceptible to different text styles**.

DFKI Challenge Set The submission by DFKI (Avramidis and Macketanz, 2022) employs a linguistically motivated challenge set that includes about 20,000 items extracted from 145 MT systems for two language directions (German⇔English). It is based on a test suite (Macketanz et al., 2022) that covers more than 100 linguistically-motivated phenomena organized in 14 categories.

The best performing metrics are YiSi-1, BERTScore and COMET-22 for German-English, and UniTE, UniTE-ref, MetricX-xl-DA-2019 and MetricX-xxl-DA-2019 for English-German. Metrics in both directions are performing worst when it comes to **named-entities & terminology** and particularly **measuring units**. Particularly in German-English they are weak at detecting issues at **punctuation, polar questions, relative clauses, dates** and **idioms**. In English-German, they perform worst at **present progressive of transitive verbs, future II progressive of intransitive verbs, simple present perfect of ditransitive verbs** and **focus particles**.

9 Conclusion

This paper summarizes the results of the WMT22 shared task on automated machine translation evaluation, the Metrics Shared Task. We presented an extensive analysis on how well metrics perform on our three main language pairs: English→German, English→Russian and Chinese→English. The results, based on 201 different tasks, demonstrated the superiority of neural-based learned metrics over overlap-based metrics like Bleu, spBleu or chrF. These results are confirmed with DA+SQM human judgement. Although this was already the case in the previous years' Metric Shared Tasks, we further strengthened the case for neural-based fine-tuned metrics by demonstrating their superiority across four different domains. In addition, we initiated a challenge set subtask, where participants had to create contrastive test suites for evaluating metrics' ability to capture and penalise specific types of translation errors.

10 Ethical Considerations

MQM annotations and additional reference translations in this paper are done by professional translators. They are all paid at professional rates.

Organizers from the National Research Council

Canada and Unbabel have submitted to this task the frozen stable versions of their metrics (YiSi and COMET) dated before this year's shared task and publicly available. Newer versions of COMET were developed without using any of the test set, test suite or challenge sets.

11 Acknowledgments

Results for this shared task would not be possible without tight collaboration with the organizers of the General MT Task. We are grateful to Google and Unbabel for sponsoring and overseeing the human evaluation.

Ricardo Rei and André F. T. Martins are supported by the P2020 program (MAIA: contract 045909) and by European Union's Horizon Europe Research and Innovation Actions (UTTER: contract 101070631). Additionally, André F. T. Martins is supported by the European Research Council (ERC StG DeepSPIN, 758969) and Fundação para a Ciência e Tecnologia (contract UIDB/50008/2020).

Eleftherios Avramidis is supported by the German Ministry of Education and Research through the research program SocialWear (grant num. 01IW2000).

References

Duarte M. Alves, Ricardo Rei, Ana C. Farinha, José G. C. de Souza, and André F. T. Martins. 2022. Robust MT evaluation with Sentence-level Multilingual data Augmentation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Chantal Amrhein, Nikita Moghe, and Liane K. Guillou. 2022. ACES: Translation Accuracy Challenge Sets for Evaluating Machine Translation Metrics. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Eleftherios Avramidis and Vivien Macketanz. 2022. Linguistically motivated evaluation of machine translation metrics based on a challenge set. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Xiaoyu Chen, Daimeng Wei, Hengchao Shang, Zongyao Li, Zhanglin Wu, Zhengzhe Yu, Ting Zhu, Mengli Zhu, Ning Xie, Lizhi Lei, Shimin Tao, Hao Yang, and Ying Qin. 2022. Exploring Robustness of Machine Translation Metrics: A Study of Twenty-Two Automatic Metrics in the WMT22 Metric Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Pierre Colombo, Nathan Noiry, Ekhine Irurozki, and Stéphan Clémençon. 2022. What are the best systems? new perspectives on nlp benchmarking. *arXiv preprint arXiv:2202.03799*.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2021. A statistical analysis of summarization evaluation metrics using resampling methods. *arXiv preprint arXiv:2104.00054*.

Bryan Eikema and Wilker Aziz. 2020. Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Bryan Eikema and Wilker Aziz. 2021. Sampling-based minimum bayes risk decoding for neural machine translation.

Allyson Ettinger, Sudha Rao, Hal Daumé III, and Emily M. Bender. 2017. Towards linguistically generalizable NLP systems: A workshop and shared task. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 1–10, Copenhagen, Denmark. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(1).

Patrick Fernandes, António Farinhas, Ricardo Rei, José De Souza, Perez Ogayo, Graham Neubig, and Andre Martins. 2022. Quality-aware decoding for neural machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1396–1412, Seattle, United States. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021a. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. 2022. High quality rather than high model probability: Minimum Bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021b. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, Maja Popović, and Mariya Shmatova. 2022a. Findings of the 2022 conference on machine translation (wmt22). In *Proceedings of the Seventh Conference on Machine Translation*. Association for Computational Linguistics.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation.

Tom Kocmi, Hitokazu Matsushita, and Christian Federmann. 2022b. MS-COMET: Larger Filtered Human Annotations Help Metric Performance. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Yilun Liu, Xiaosong Qiao, Zhanglin Wu, Su Chang, Min Zhang, Yanqing Zhao, shimin tao Song Peng, Hao Yang, Ying Qin, Jiaxin Guo, Minghan Wang, Yinglu Li, Peng Li, and Xiaofeng Zhao. 2022. Partial Could Be Better Than Whole: HW-TSC 2022 Submission for the Metrics Shared Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Arle Lommel, Hans Uszkoreit, and Aljoscha Burchardt. 2014. Multidimensional Quality Metrics (MQM) : A Framework for Declaring and Describing Translation Quality Metrics. *Tradumàtica*, pages 0455–463.

Vivien Macketanz, Eleftherios Avramidis, Aljoscha Burchardt, He Wang, Renlong Ai, Shushen Manakhimova, Ursula Strohriegel, Sebastian Möller, and Hans Uszkoreit. 2022. A linguistically motivated test suite to semi-automatically evaluate German–English machine translation output. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 936–947, Marseille, France. European Language Resources Association.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in bleu: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997.

Ananya Mukherjee, Hema Ala, Manish Shrivastava, and Dipti Misra Sharma. 2020. Mee: an automatic metric for evaluation using embeddings for machine translation. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 292–299. IEEE.

Ananya Mukherjee and Manish Shrivastava. 2022a. REUSE: REference-free UnSupervised quality Estimation Metric. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Ananya Mukherjee and Manish Shrivastava. 2022b. Unsupervised Embedding-based Metric for MT Evaluation with Improved Human Correlation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Mathias Müller and Rico Sennrich. 2021. Understanding the properties of minimum Bayes risk decoding in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 259–272, Online. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No Language Left Behind: Scaling Human-Centered Machine Translation.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Stefano Perrella, Lorenzo Proietti, Alessandro Scirè, Niccolò Campolungo, and Roberto Navigli. 2022. Machine Translation Evaluation as a Sequence Tagging Problem. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Amy Pu, Hyung Won Chung, Ankur Parikh, Sebastian Gehrmann, and Thibault Sellam. 2021. Learning compact metrics for MT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 751–762, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 Submission for the Metrics Shared Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Ricardo Rei, Ana C Farinha, Chrysoula Zerva, Daan van Stigt, Craig Stewart, Pedro Ramos, Taisiya Glushkova, André F. T. Martins, and Alon Lavie. 2021. Are references really needed? unbabel-IST 2021 submission for the metrics shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1030–1040, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online. Association for Computational Linguistics.

Yu Wan, Keqin Bao, Dayiheng Liu, Baosong Yang, Derek F. Wong, Lidia S. Chao, Wenqiang Lei, and Jun Xie. 2022a. Alibaba-Translate China's Submission for WMT2022 Metrics Shared Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Yu Wan, Dayiheng Liu, Baosong Yang, Haibo Zhang, Boxing Chen, Derek Wong, and Lidia Chao. 2022b. UniTE: Unified translation evaluation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8117–8127, Dublin, Ireland. Association for Computational Linguistics.

Wenda Xu, Yilin Tuan, Yujie Lu, Michael Saxon, Lei Li, and William Yang Wang. 2022. Not all errors are equal: Learning text generation metrics using stratified error synthesis. *arXiv preprint arXiv:2210.05035*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Task	Accuracy	en-de	en-de	en-ru	zh-en	zh-en
Human Translation Included	No	Yes	No	No	Yes	No
metricx_xl_DA_2019	0.865	0.908	0.905	0.977	0.966	0.982
metricx_xxl_DA_2019	0.865	0.907	0.901	0.982	0.961	0.984
metricx_xxl_MQM_2020	0.850	0.862	0.847	0.949	0.924	0.920
BLEURT-20	0.847	0.691	0.719	0.959	0.909	0.938
metricx_xl_MQM_2020	0.843	0.848	0.832	0.927	0.920	0.914
COMET-22	0.839	0.761	0.771	0.900	0.947	0.942
COMET-20	0.836	0.812	0.876	0.936	0.964	0.970
UniTE	0.828	0.642	0.624	0.888	0.922	0.914
MS-COMET-22	0.828	0.634	0.695	0.809	0.918	0.909
UniTE-ref	0.818	0.652	0.632	0.831	0.902	0.892
MATESE	0.810	0.647	0.617	0.757	0.869	0.856
YiSi-1	0.792	0.506	0.626	0.881	0.867	0.935
MEE4	0.788	0.404	0.537	0.792	0.818	0.905
COMETKiwi*	0.788	0.592	0.674	0.763	0.795	0.866
HuaweiTSC_EE_BERTScore_0.8_With_Human	0.785	0.354	0.463	0.818	0.903	0.960
HuaweiTSC_EE_BERTScore_0.8_Without_Human	0.785	0.338	0.451	0.818	0.900	0.957
Cross-QE*	0.781	0.643	0.661	0.806	0.817	0.870
HuaweiTSC_EE_BERTScore_0.5_With_Human	0.781	0.287	0.400	0.792	0.938	0.953
COMET-QE*	0.781	0.480	0.502	0.468	0.544	0.569
HuaweiTSC_EE_BERTScore_0.5_Without_Human	0.774	0.246	0.370	0.795	0.930	0.942
BERTScore	0.774	0.338	0.428	0.811	0.843	0.924
HuaweiTSC_EE_BERTScore_0.3_With_Human	0.759	0.243	0.356	0.754	0.945	0.943
UniTE-src*	0.759	0.509	0.509	0.779	0.791	0.874
MEE2	0.759	0.360	0.479	0.811	0.753	0.872
MS-COMET-QE-22*	0.755	0.417	0.539	0.672	0.799	0.897
MATESE-QE*	0.748	0.363	0.337	0.637	0.741	0.767
MEE	0.748	0.358	0.445	0.823	0.727	0.824
f101spBLEU	0.745	0.210	0.298	0.816	0.613	0.718
f200spBLEU	0.741	0.230	0.283	0.819	0.614	0.728
HuaweiTSC_EE_BERTScore_0.3_Without_Human	0.737	0.189	0.316	0.761	0.931	0.926
chrF	0.734	0.159	0.346	0.815	0.647	0.630
BLEU	0.708	0.038	0.179	0.724	0.579	0.594
HWTSC-TLM*	0.697	0.311	0.428	0.597	0.368	0.460
HWTSC-Teacher-Sim*	0.686	0.290	0.385	0.675	0.294	0.356
KG-BERTScore*	0.664	0.369	0.400	0.612	0.617	0.743
REUSE*	0.347	-0.514	-0.465	-0.349	-0.330	-0.142
SEScore	–	0.581	0.660	–	0.920	0.944

Table 10: Pearson correlation of all metrics with system-level MQM scores for the three main language pairs. Rows are sorted by the system-level pairwise accuracy across the three language pairs. Primary submissions are bolded, and baselines are underlined. Reference-free metrics are indicated using an asterisk.

A Language-Specific Results Tables

Language-specific results are given in Table 10 and Table 11. Each page contains results for scores over all domains over a single granularity (system or segment).

For all tables, the correlations are calculated on metric scores comparing MT system translations with Reference A, and any additional human reference translations are not included.

For segment level correlation, we report results on the "none" averaging method, where we flatten the matrices into single vectors before computing the Kendall Tau correlation.

B Correlations with WMT Human Evaluation

Correlations with WMT Direct Assessment Human scores are given in the following tables, with results for language pairs evaluated using reference-based Direct Assessment (Ref. DA) (Graham et al., 2013), followed by results for language pairs evaluated using SQM style source-based DA (DA+SQM) (Kocmi et al., 2022a). Since most language pairs contained only a single reference, we used reference A for all pairs, and report results only for scoring MT output (omitting additional scored references for language pairs where these were available). System-level correlations use Pearson and segment-level scores use Kendall. For simplicity, both statistics are computed over raw rater scores, with no traditional difference-25

Task	(sys) Accuracy	en-de	en-de	en-ru	zh-en	zh-en
Human Translation Included	No	Yes	No	No	Yes	No
metricx_xl_DA_2019	0.865	0.356	0.362	0.393	0.383	0.392
metricx_xxl_DA_2019	0.865	0.355	0.361	0.405	0.377	0.386
metricx_xxl_MQM_2020	0.850	0.356	0.360	0.420	0.421	0.427
BLEURT-20	0.847	0.338	0.344	0.359	0.352	0.361
metricx_xl_MQM_2020	0.843	0.362	0.367	0.383	0.416	0.423
COMET-22	0.839	0.361	0.368	0.400	0.420	0.428
COMET-20	0.836	0.312	0.319	0.330	0.325	0.332
UniTE	0.828	0.362	0.369	0.378	0.351	0.357
MS-COMET-22	0.828	0.277	0.283	0.351	0.335	0.341
UniTE-ref	0.818	0.356	0.362	0.374	0.354	0.361
MATESE	0.810	0.323	0.323	0.279	0.382	0.389
YiSi-1	0.792	0.229	0.235	0.227	0.288	0.296
MEE4	0.788	0.236	0.243	0.210	0.189	0.194
COMETKiwi*	0.788	0.283	0.290	0.359	0.352	0.364
HuaweiTSC_EE_BERTScore_0.8_With_Human	0.785	–	–	–	–	–
HuaweiTSC_EE_BERTScore_0.8_Without_Human	0.785	–	–	–	–	–
Cross-QE*	0.781	0.259	0.263	0.310	0.368	0.378
HuaweiTSC_EE_BERTScore_0.5_With_Human	0.781	–	–	–	–	–
COMET-QE*	0.781	0.277	0.281	0.341	0.356	0.365
HuaweiTSC_EE_BERTScore_0.5_Without_Human	0.774	–	–	–	–	–
BERTScore	0.774	0.226	0.232	0.192	0.307	0.316
HuaweiTSC_EE_BERTScore_0.3_With_Human	0.759	–	–	–	–	–
UniTE-src*	0.759	0.283	0.287	0.342	0.332	0.343
MEE2	0.759	0.238	0.244	0.201	0.197	0.201
MS-COMET-QE-22*	0.755	0.226	0.233	0.305	0.277	0.287
MATESE-QE*	0.748	0.242	0.244	0.229	0.328	0.337
MEE	0.748	0.187	0.192	0.148	0.149	0.149
f101spBLEU	0.745	0.169	0.174	0.135	0.143	0.145
f200spBLEU	0.741	0.176	0.180	0.153	0.139	0.140
HuaweiTSC_EE_BERTScore_0.3_Without_Human	0.737	–	–	–	–	–
chrF	0.734	0.208	0.214	0.168	0.146	0.147
BLEU	0.708	0.164	0.169	0.140	0.143	0.145
HWTSC-TLM*	0.697	0.087	0.092	0.121	0.079	0.086
HWTSC-Teacher-Sim*	0.686	0.150	0.155	0.143	0.264	0.272
KG-BERTScore*	0.664	0.126	0.129	0.111	0.214	0.219
REUSE*	0.347	0.057	0.065	0.078	0.116	0.130
SEScore	–	0.261	0.266	–	0.324	0.331

Table 11: Kendall Tau correlation of all metrics with segment-level MQM scores for the three main language pairs. Rows are sorted by the system-level pairwise accuracy across the three language pairs. Primary submissions are bolded, and baselines are underlined. Reference-free metrics are indicated using an asterisk.

filtering.[12]

[12] The traditional recipe made little difference in overall correlation patterns.

Task	Accuracy	cs-en	de-en	ja-en	ru-en	uk-en	zh-en
Incl. Human Translation	False	False	False	False	False	False	False
f200spBLEU	0.669	0.812	0.405	0.949	0.831	0.714	0.517
chrF	0.666	0.806	0.354	0.983	0.827	0.688	0.568
BERTScore	0.666	0.825	0.440	0.988	0.851	0.717	0.396
YiSi-1	0.660	0.824	0.443	0.989	0.847	0.708	0.415
f101spBLEU	0.660	0.810	0.406	0.944	0.830	0.718	0.521
BLEU	0.653	0.801	0.352	0.934	0.843	0.648	0.563
BLEURT-20	0.650	0.833	0.458	0.990	0.849	0.733	0.266
HWTSC_EE_BERTScore_0.8_Without_Human	0.647	0.824	0.442	0.989	0.858	0.714	0.417
HWTSC_EE_BERTScore_0.3_Without_Human	0.647	0.808	0.391	0.987	0.876	0.678	0.437
HWTSC_EE_BERTScore_0.3_With_Human	0.647	0.799	0.390	0.987	0.876	0.680	0.412
HWTSC_EE_BERTScore_0.8_With_Human	0.644	0.820	0.440	0.989	0.858	0.715	0.411
HWTSC_EE_BERTScore_0.5_With_Human	0.644	0.808	0.410	0.988	0.870	0.696	0.416
HWTSC_EE_BERTScore_0.5_Without_Human	0.644	0.815	0.411	0.988	0.870	0.695	0.434
MS-COMET-QE-22*	0.641	0.769	0.395	0.990	0.867	0.699	0.312
COMET-20	0.635	0.827	0.424	0.989	0.847	0.723	0.330
metricx_xxl_DA_2019	0.635	0.831	0.469	0.987	0.850	0.730	0.148
UniTE-ref	0.629	0.822	0.440	0.982	0.855	0.727	0.167
MS-COMET-22	0.626	0.807	0.419	0.990	0.858	0.701	0.108
COMET-22	0.626	0.821	0.446	0.976	0.857	0.714	0.135
metricx_xl_DA_2019	0.623	0.833	0.468	0.987	0.851	0.730	0.157
Cross-QE*	0.623	0.791	0.415	0.989	0.863	0.719	0.129
UniTE	0.623	0.832	0.431	0.984	0.852	0.728	0.195
UniTE-src*	0.623	0.777	0.402	0.989	0.863	0.703	0.210
metricx_xl_MQM_2020	0.620	0.821	0.487	0.978	0.856	0.718	-0.039
metricx_xxl_MQM_2020	0.620	0.823	0.490	0.978	0.856	0.715	-0.061
COMETKiwi*	0.617	0.787	0.409	0.984	0.862	0.718	0.181
COMET-QE*	0.605	0.811	0.443	0.981	0.864	0.744	-0.006
REUSE*	0.584	0.200	0.194	0.990	0.683	0.150	0.531
HWTSC-TLM*	0.578	0.822	0.356	0.980	0.842	0.695	0.083
HWTSC-Teacher-Sim*	0.568	0.804	0.322	0.985	0.848	0.691	-0.011
KG-BERTScore*	0.568	0.539	0.052	0.989	0.805	0.516	0.264
MEE	–	–	–	–	–	–	0.578
MEE2	–	–	–	–	–	–	0.511
MEE4	–	–	–	–	–	–	0.455
SEScore	–	–	–	–	–	–	0.331
MATESE	–	–	–	–	–	–	0.013
MATESE-QE*	–	–	–	–	–	–	0.013

Table 12: System-level Pearson correlation with crowdsourced Ref. DA scores. Rows are sorted by the system-level pairwise accuracy across all language pairs. Primary submissions are bolded, and baselines are underlined. Reference-free metrics are indicated using an asterisk.

System-level Metric accuracy and correlations with REFDA scores contradict the main results. We strongly recommend against using Ref. DA scores to evaluate MT metrics.

Task	(sys) Accuracy	cs-en	de-en	ja-en	ru-en	uk-en	zh-en
Incl. Human Translation	False	False	False	False	False	False	False
f200spBLEU	0.669	0.043	0.010	0.085	0.018	0.006	0.026
chrF	0.666	0.042	0.017	0.083	0.015	0.003	0.025
BERTScore	0.666	0.039	0.011	0.084	0.019	0.003	0.020
YiSi-1	0.660	0.037	0.012	0.087	0.018	0.004	0.020
f101spBLEU	0.660	0.042	0.010	0.085	0.020	0.008	0.026
BLEU	0.653	0.043	0.009	0.081	0.014	0.007	0.024
BLEURT-20	0.650	0.036	0.018	0.085	0.014	0.002	0.013
HWTSC_EE_BERTScore_0.8_Without_Human	0.647	–	–	–	–	–	–
HWTSC_EE_BERTScore_0.3_Without_Human	0.647	–	–	–	–	–	–
HWTSC_EE_BERTScore_0.3_With_Human	0.647	–	–	–	–	–	–
HWTSC_EE_BERTScore_0.8_With_Human	0.644	–	–	–	–	–	–
HWTSC_EE_BERTScore_0.5_With_Human	0.644	–	–	–	–	–	–
HWTSC_EE_BERTScore_0.5_Without_Human	0.644	–	–	–	–	–	–
MS-COMET-QE-22*	0.641	0.022	0.011	0.088	-0.002	0.003	0.001
COMET-20	0.635	0.034	0.018	0.084	0.014	-0.002	0.009
metricx_xxl_DA_2019	0.635	0.040	0.019	0.086	0.015	0.005	0.008
UniTE-ref	0.629	0.032	0.018	0.084	0.009	0.004	0.005
MS-COMET-22	0.626	0.030	0.013	0.081	0.007	-0.000	0.004
COMET-22	0.626	0.031	0.019	0.079	0.013	0.002	0.002
metricx_xl_DA_2019	0.623	0.036	0.016	0.085	0.014	0.002	0.007
Cross-QE*	0.623	0.015	0.011	0.087	0.003	0.001	-0.000
UniTE	0.623	0.036	0.019	0.084	0.012	0.004	0.006
UniTE-src*	0.623	0.026	0.018	0.087	0.001	0.003	0.007
metricx_xl_MQM_2020	0.620	0.025	0.013	0.079	0.010	0.004	-0.002
metricx_xxl_MQM_2020	0.620	0.026	0.014	0.079	0.011	0.002	-0.003
COMETKiwi*	0.617	0.028	0.011	0.091	0.001	0.004	0.002
COMET-QE*	0.605	0.010	0.020	0.076	-0.005	-0.002	0.003
REUSE*	0.584	0.002	0.009	0.091	-0.007	0.000	0.011
HWTSC-TLM*	0.578	0.030	0.011	0.097	0.013	0.001	0.013
HWTSC-Teacher-Sim*	0.568	0.018	0.016	0.098	0.007	0.007	0.001
KG-BERTScore*	0.568	0.010	0.007	0.087	-0.012	0.008	-0.002
MEE	–	–	–	–	–	–	0.020
MEE2	–	–	–	–	–	–	0.021
MEE4	–	–	–	–	–	–	0.021
SEScore	–	–	–	–	–	–	0.013
MATESE	–	–	–	–	–	–	-0.009
MATESE-QE*	–	–	–	–	–	–	-0.006

Table 13: Segment-level Kendall-like correlation with crowdsourced Ref. DA scores. Rows are sorted by the system-level pairwise accuracy across all language pairs. Primary submissions are bolded, and baselines are underlined. Reference-free metrics are indicated using an asterisk.

The segment level Kendal-like correlations of all metrics with Ref. DA scores are all very close to zero, and these numbers are completely meaningless. We strongly recommend against using Ref. DA scores to evaluate MT metrics.

Task Incl. Human Translation	accuracy True	cs-uk False	en-cs False	en-de False	en-hr False	en-ja False	en-liv False	en-ru False	en-uk False	en-zh False	liv-en False	sah-ru False	uk-cs False	zh-en False
metricx_xl_MQM_2020	0.862	0.989	0.833	0.674	0.981	0.885	0.819	0.950	0.916	0.809	0.973	1.000	0.964	0.829
metricx_xxl_MQM_2020	0.862	0.989	0.853	0.713	0.961	0.885	0.913	0.963	0.939	0.838	0.907	1.000	0.961	0.807
metricx_xxl_DA_2019	0.860	0.984	0.861	0.748	0.972	0.913	0.936	0.954	0.943	0.841	0.994	1.000	0.944	0.887
metricx_xl_DA_2019	0.853	0.985	0.837	0.731	0.979	0.920	0.931	0.937	0.926	0.813	0.997	1.000	0.942	0.903
UniTE	0.849	0.990	0.837	0.514	0.985	0.923	0.905	0.930	0.919	0.811	0.999	1.000	0.948	0.890
COMET-22	0.842	0.987	0.831	0.593	0.939	0.902	0.922	0.922	0.913	0.765	0.994	1.000	0.959	0.893
UniTE-ref	0.840	0.989	0.849	0.523	0.978	0.918	0.896	0.937	0.921	0.814	1.000	1.000	0.944	0.877
Cross-QE*	0.835	0.976	0.792	0.614	0.966	0.904	-0.395	0.936	0.910	0.714	0.984	1.000	0.951	0.835
COMETKiwi*	0.835	0.976	0.832	0.525	0.931	0.923	0.616	0.867	0.902	0.761	0.993	1.000	0.962	0.876
MS-COMET-22	0.833	0.978	0.763	0.265	0.940	0.927	0.942	0.906	0.785	0.822	0.999	1.000	0.962	0.856
BLEURT-20	0.830	0.989	0.832	0.707	0.973	0.907	0.956	0.931	0.937	0.665	0.998	1.000	0.951	0.906
COMET-20	0.826	0.985	0.739	0.626	0.974	0.915	0.957	0.914	0.910	0.744	0.998	1.000	0.953	0.913
MS-COMET-QE-22*	0.824	0.965	0.682	-0.047	0.905	0.940	0.911	0.822	0.702	0.822	0.999	1.000	0.953	0.833
COMET-QE*	0.821	0.922	0.828	0.522	0.781	0.881	0.015	0.941	0.881	0.709	0.998	1.000	0.921	0.818
UniTE-src*	0.800	0.955	0.765	0.396	0.966	0.921	-0.225	0.913	0.892	0.752	0.993	1.000	0.957	0.829
YiSi-1	0.785	0.960	0.632	0.747	0.921	0.929	0.986	0.804	0.889	0.509	0.987	1.000	0.950	0.932
HWTSC_EE_BERTScore_0.8_With_Human	0.780	0.938	0.489	0.674	0.860	0.921	0.958	0.703	0.828	0.612	0.997	1.000	0.960	0.950
HWTSC_EE_BERTScore_0.8_Without_Human	0.777	0.937	0.511	0.669	0.843	0.923	0.957	0.703	0.828	0.584	0.997	1.000	0.960	0.944
HWTSC_EE_BERTScore_0.5_With_Human	0.771	0.943	0.361	0.625	0.867	0.893	0.960	0.686	0.826	0.734	0.990	1.000	0.935	0.949
HWTSC_EE_BERTScore_0.5_Without_Human	0.766	0.943	0.438	0.609	0.821	0.903	0.960	0.688	0.826	0.652	0.989	1.000	0.935	0.933
BERTScore	0.764	0.935	0.482	0.648	0.890	0.932	0.969	0.702	0.825	0.412	0.993	1.000	0.965	0.937
chrF	0.762	0.927	0.689	0.811	0.920	0.931	0.969	0.813	0.895	0.210	0.988	1.000	0.979	0.881
HWTSC_EE_BERTScore_0.3_With_Human	0.750	0.933	0.203	0.552	0.869	0.857	0.961	0.655	0.793	0.774	0.984	1.000	0.908	0.931
HWTSC-TLM*	0.748	0.880	0.811	0.001	0.574	0.837	0.428	0.821	0.578	0.667	0.947	1.000	0.913	0.307
f101spBLEU	0.748	0.883	0.567	0.690	0.901	0.866	0.991	0.698	0.825	0.197	0.983	1.000	0.976	0.886
f200spBLEU	0.748	0.888	0.549	0.656	0.906	0.862	0.989	0.690	0.814	0.208	0.979	1.000	0.974	0.891
HWTSC_EE_BERTScore_0.3_Without_Human	0.732	0.935	0.327	0.521	0.806	0.875	0.961	0.659	0.794	0.678	0.982	1.000	0.909	0.910
HWTSC-Teacher-Sim*	0.720	0.850	0.516	-0.072	0.839	0.842	0.706	0.580	0.499	0.591	0.900	1.000	0.941	0.363
BLEU	0.707	0.890	0.666	0.493	0.919	0.282	0.804	0.649	0.752	0.065	0.979	1.000	0.982	0.859
KG-BERTScore*	0.484	0.366	-0.845	-0.128	0.331	0.065	-0.734	0.250	-0.834	-0.123	0.458	-1.000	0.900	0.466
REUSE*	0.344	0.042	-0.923	-0.409	0.223	-0.096	-0.713	-0.859	-0.873	-0.202	0.644	-1.000	0.653	-0.158
MATESE	–	–	–	0.460	–	–	–	0.891	–	–	–	–	–	0.843
MEE	–	–	–	0.746	–	–	–	0.767	–	–	–	–	–	0.823
MEE2	–	–	–	0.774	–	–	–	0.765	–	–	–	–	–	0.876
MEE4	–	–	–	0.738	–	–	–	0.744	–	–	–	–	–	0.897
MATESE-QE*	–	–	–	0.151	–	–	–	0.876	–	–	–	–	–	0.846
SEScore	–	–	–	0.138	–	–	–	–	–	–	–	–	–	0.924

Table 14: System-level Pearson correlation with WMT source-based DA+SQM scores. Rows are sorted by the system-level pairwise accuracy across all language pairs. Primary submissions are bolded, and baselines are underlined. Reference-free metrics are indicated using an asterisk.

Task Incl. Human Translation	(sys) Accuracy True	cs-uk False	en-cs False	en-de False	en-hr False	en-ja False	en-liv False	en-ru False	en-uk False	en-zh False	liv-en False	sah-ru False	uk-cs False	zh-en False
metricx_xl_MQM_2020	0.862	0.276	0.273	0.133	0.289	0.218	0.006	0.288	0.285	0.141	0.147	0.384	0.266	0.260
metricx_xxl_MQM_2020	0.862	0.295	0.306	0.143	0.310	0.242	0.021	0.312	0.315	0.156	0.108	0.382	0.288	0.268
metricx_xxl_DA_2019	0.860	0.315	0.299	0.152	0.318	0.239	0.109	0.317	0.331	0.149	0.208	0.500	0.305	0.244
metricx_xl_DA_2019	0.853	0.306	0.299	0.143	0.306	0.237	0.126	0.308	0.322	0.148	0.218	0.503	0.296	0.251
UniTE	0.849	0.311	0.314	0.135	0.320	0.256	0.107	0.284	0.335	0.148	0.230	0.515	0.305	0.211
COMET-22	0.842	0.309	0.317	0.127	0.316	0.230	0.078	0.303	0.328	0.156	0.186	0.470	0.326	0.271
UniTE-ref	0.840	0.315	0.315	0.131	0.315	0.253	0.094	0.280	0.338	0.149	0.235	0.517	0.308	0.210
Cross-QE*	0.835	0.232	0.267	0.085	0.208	0.241	-0.075	0.225	0.232	0.137	0.137	0.300	0.238	0.254
COMETKiwi*	0.835	0.288	0.295	0.111	0.255	0.202	0.017	0.255	0.299	0.129	0.173	0.359	0.287	0.231
MS-COMET-22	0.833	0.276	0.298	0.114	0.292	0.235	0.108	0.281	0.307	0.141	0.214	0.445	0.253	0.238
BLEURT-20	0.830	0.292	0.291	0.140	0.274	0.221	0.091	0.283	0.317	0.133	0.227	0.497	0.276	0.218
COMET-20	0.826	0.280	0.279	0.133	0.298	0.244	0.135	0.280	0.297	0.141	0.237	0.488	0.270	0.214
MS-COMET-QE-22*	0.824	0.247	0.245	0.093	0.261	0.179	0.151	0.249	0.260	0.127	0.167	0.341	0.214	0.220
COMET-QE*	0.821	0.225	0.258	0.089	0.249	0.181	-0.050	0.240	0.265	0.119	0.125	0.235	0.238	0.232
UniTE-src*	0.800	0.283	0.301	0.116	0.293	0.253	-0.015	0.256	0.322	0.150	0.179	0.314	0.288	0.215
YiSi-1	0.785	0.223	0.173	0.084	0.225	0.212	0.120	0.191	0.211	0.076	0.212	0.439	0.203	0.168
HWTSC_EE_BERTScore_0.8_With_Human	0.780	–	–	–	–	–	–	–	–	–	–	–	–	–
HWTSC_EE_BERTScore_0.8_Without_Human	0.777	–	–	–	–	–	–	–	–	–	–	–	–	–
HWTSC_EE_BERTScore_0.5_With_Human	0.771	–	–	–	–	–	–	–	–	–	–	–	–	–
HWTSC_EE_BERTScore_0.5_Without_Human	0.766	–	–	–	–	–	–	–	–	–	–	–	–	–
BERTScore	0.764	0.200	0.165	0.091	0.215	0.177	0.119	0.173	0.177	0.072	0.217	0.438	0.190	0.188
chrF	0.762	0.195	0.147	0.085	0.185	0.142	0.101	0.153	0.177	0.051	0.184	0.430	0.171	0.071
HWTSC_EE_BERTScore_0.3_With_Human	0.750	–	–	–	–	–	–	–	–	–	–	–	–	–
HWTSC-TLM*	0.748	0.122	0.059	0.035	0.076	0.081	0.051	0.102	0.098	0.023	0.100	0.105	0.062	0.039
f101spBLEU	0.748	0.154	0.131	0.070	0.179	0.131	0.098	0.124	0.145	0.049	0.146	0.372	0.162	0.074
f200spBLEU	0.748	0.160	0.133	0.069	0.176	0.133	0.089	0.132	0.155	0.050	0.148	0.383	0.162	0.069
HWTSC_EE_BERTScore_0.3_Without_Human	0.732	–	–	–	–	–	–	–	–	–	–	–	–	–
HWTSC-Teacher-Sim*	0.720	0.116	0.115	0.049	0.148	0.119	0.032	0.104	0.177	0.056	0.068	0.054	0.090	0.168
BLEU	0.707	0.133	0.136	0.069	0.179	0.038	0.036	0.122	0.150	0.032	0.139	0.361	0.150	0.077
KG-BERTScore*	0.484	0.175	0.085	0.040	0.154	0.082	-0.098	0.087	0.121	0.054	0.016	0.013	0.137	0.149
REUSE*	0.344	0.155	0.049	0.031	0.079	0.107	-0.110	0.049	0.101	0.081	0.032	0.108	0.115	0.093
MATESE	–	–	–	0.106	–	–	–	0.198	–	–	–	–	–	0.225
MEE	–	–	–	0.071	–	–	–	0.118	–	–	–	–	–	0.078
MEE2	–	–	–	0.092	–	–	–	0.171	–	–	–	–	–	0.117
MEE4	–	–	–	0.091	–	–	–	0.181	–	–	–	–	–	0.104
MATESE-QE*	–	–	–	0.083	–	–	–	0.173	–	–	–	–	–	0.220
SEScore	–	–	–	0.091	–	–	–	–	–	–	–	–	–	0.213

Table 15: Segment-level Kendall-like correlation with DA+SQM scores. Rows are sorted by the system-level pairwise accuracy across all language pairs. Primary submissions are bolded, and baselines are underlined. Reference-free metrics are indicated using an asterisk.

Findings of the WMT 2022 Shared Task on Quality Estimation

Chrysoula Zerva[1,2], Frédéric Blain[3], Ricardo Rei[2,4,5], Piyawat Lertvittayakumjorn[6], José G. C. de Souza[4], Steffen Eger[9], Diptesh Kanojia[8], Duarte Alves[2], Constantin Orăsan[8], Marina Fomicheva[7], André F. T. Martins[1,2,4] and Lucia Specia[6,7]

[1]Instituto de Telecomunicações, [2]Instituto Superior Técnico, [3]University of Wolverhampton
[4]Unbabel, [5]INESC-ID, [6]Imperial College London, [7]University of Sheffield
[8]University of Surrey, [9]NLLG,Technische Fakultät, Bielefeld University

f.blain@wlv.ac.uk,m.fomicheva@sheffield.ac.uk, {d.kanojia, c.orasan}@surrey.ac.uk
{chrysoula.zerva,ricardo.rei,duartemalves}@tecnico.ulisboa.pt, pl1515@ic.ac.uk

Abstract

We report the results of the WMT 2022 shared task on Quality Estimation, in which the challenge is to predict the quality of the output of neural machine translation systems at the word and sentence levels, without access to reference translations. This edition introduces a few novel aspects and extensions that aim to enable more fine-grained, and explainable quality estimation approaches. We introduce an updated quality annotation scheme using Multidimensional Quality Metrics to obtain sentence- and word-level quality scores for three language pairs. We also extend the Direct Assessments and post-edit data (MLQE-PE) to new language pairs: we present a novel and large dataset on English-Marathi, as well as a zero-shot test-set on English-Yoruba. Further, we include an explainability sub-task for all language pairs and present a new format of a critical error detection task for two new language pairs. Participants from 11 different teams submitted altogether 991 systems to different task variants and language pairs.

1 Introduction

The 11th edition of the shared task on Quality Estimation (QE) builds on its previous editions and findings to further benchmark methods for estimating the quality of neural machine translation (MT) output at run-time, without the use of reference translations. It includes (sub)tasks that consider quality of machine translations at the word and sentence levels.

Over the past years, the QE field has been moving towards trainable, large, multilingual models that have been shown to achieve high performance, especially at sentence-level (Specia et al., 2021). In this edition, we further expand the provided resources, introducing new low-resource language pairs: a large dataset of English-Marathi, suitable for training, development and testing and a smaller test-set on English-Yoruba for zero-shot approaches. These, as well as previously published datasets for QE, rely mainly on Direct Assessments (DA)[1] and post-edited translations, which provide estimates of quality either by using the human quality score(s) for each segment or by estimating the distance of a translation from a human-provided correction. As these annotations can sometimes obscure the exact location and/or significance of a translation error, we wanted to investigate the feasibility and efficiency of using a more fine-grained annotation schema to obtain quality estimations at word- and sentence- level, namely Multidimensional Quality Metrics (MQM) (Lommel et al., 2014). MQM annotations have shown to be more trustworthy for the metrics task (Freitag et al., 2021a,b), motivating us to evaluate their suitability for the QE task. We make available new development and test data on three language pairs using MQM annotations.

The aforementioned boost in performance of QE systems frequently comes at the cost of efficiency and interpretability, since they heavily rely on large models with many parameters. As a result, the predicted quality estimates are hard to interpret. At the same time, such high-performance, "black-box" models are frequently susceptible to systematic errors, such as negation omission (Kanojia et al., 2021) and mistranslated entities (Amrhein and Sennrich, 2022). Both phenomena are major concerns for MT quality estimation since they can undermine users' trust in new technologies and hamper the adoption of such models on a wide scale. To motivate approaches that address these cases we include an explainability subtask following its first edition at Eval4NLP 2021 (Fomicheva et al., 2021). In this subtask we ask participants to predict

[1] We note that the procedure followed for our data diverges from that proposed by Graham et al. (2016) in three ways: (a) we employ fewer but professional translators to score each sentence, (b) scoring is done against the source segment (bilingual annotation) and not the reference, and (c) we provide translators with guidelines on the meaning of ranges of scores.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 69 - 99
December 7-8, 2022 ©2022 Association for Computational Linguistics

the erroneous words as rationale extraction for a sentence-level quality estimate, without any word-level supervision. By framing error identification as rationale extraction for sentence-level quality estimation systems, this subtask offers an opportunity to study whether such systems behave in the same way as humans would do. We also reshape the critical error detection task of last year and we build a new corpus to test the ability of QE systems to detect critical errors that simulate hallucinated content with additions, deletions, named entities, polarity changes and numbers. The corpus is created using SMAUG (Alves et al., 2022) and we allow participation in constrained and unconstrained settings. For the constrained setting, participants have to build QE systems without having access to data from SMAUG, whereas participants from the unconstrained task can train their systems using additional data from SMAUG.

In addition to advancing the state-of-the-art at all prediction levels, our main goals are:

- To extend the languages covered in our datasets;
- To further motivate fine-grained quality annotation, informed at word and sentence level using MQM;
- To encourage language-independent and even unsupervised approaches especially for zero-shot prediction;
- To study and promote explainable approaches for MT evaluation; and
- To revisit critical error detection.

We thus have three tasks:

Task 1 The core QE task, consisting of separate sentence-level and word-level subtasks. For the sentence-level sub-tasks, the goal is to predict a quality score for each segment in the test set, which can be a variant of DA (§2.1.1) or MQM (§2.1.1). For the word-level subtasks, participants have to predict translation errors at word-level, via binary quality tags (see §2.1.2).

Task 2 Explainable QE task, aiming to obtain word-level rationales for sentence-level quality scores (§2.2).

Task 3 The critical Error Detection task, aiming to predict sentence-level binary scores indicating whether or not a translation contains a critical error (§2.3).

The tasks make use of large datasets annotated by professional translators with either 0-100 DA scoring, post-editing or MQM annotations. We update the training and development datasets of previous editions and provide new test sets for Tasks 1 and 2. Additionally, we provide a novel setup for Task 3, with novel train, development and test data. The datasets and models released are publicly available[2]. Participants are also allowed to explore any additional data and resources deemed relevant, across tasks.

The shared task uses CodaLab as submission platform, where participants (Section 4) could submit up to 2 submissions a day for each task and language pair (LP), up to a total of 10 submissions. Results for all tasks evaluated according to standard metrics are given in Section 5. Baseline systems were trained by the task organisers and entered in the platform to provide a basis for comparison (Section 3). A discussion on the main goals and findings from this year's task is presented in Section 6.

2 Quality Estimation tasks

In what follows, we briefly describe each subtask, including the datasets provided for them.

2.1 Task 1: Predicting translation quality

Being able to automatically predict the quality of translations on sentence- or word-level without access to human-references is the core goal of the QE shared task. In this edition, we explored some new approaches towards quality annotations for sentence- and word-level, and redefined the word-level quality labelling scheme, in an attempt to allow participants to employ multi-task approaches and exploit fine-grained quality annotations. Hence, the data was produced in two ways:

1. DA & Post-edit approach: The quality of each source-translation pair is annotated by at least 3 independent expert annotators, using DA on a scale 0-100. The translation is also post-edited to obtain the closest possible, fully correct translation of the source. Using the post-edited data, we generate Human-mediated

[2] https://github.com/WMT-QE-Task/wmt-qe-2022-data

Translation Edit Rate (HTER) (Snover et al., 2006) scores, which are obtained by calculating the minimum edit distance between the machine translation and its manually post-edited version. By additionally considering the alignment between the source and post-edited sentence, we can propagate the errors to the source sentence and annotate the segments that were potentially mistranslated and/or not translated at all. The HTER scores were made available to participants as additional data, but are not used as prediction targets.

2. MQM approach: Each source-translation pair is evaluated by at least 1 expert annotator, and errors identified in text are highlighted and classified in terms of severity (minor, major, critical) and type (omission, style, mistranslation, etc).

The DA and MQM data was further processed to a) obtain normalised quality scores that have the same direction between high and low quality and b) obtain word-level binary quality labels. We provide more details on the required pre-processing in §2.1.1 and §2.1.2.

DA & Post-edit data: For all language pairs the data provided is selected from publicly available resources. Specifically for training we used the following language pairs from the MLQE-PE dataset (Fomicheva et al., 2022): English-German (En-De), English-Chinese (En-Zh), Russian-English (Ru-En), Romanian-English (Ro-En), Nepalese-English (Ne-En), Esthonian-English (Et-En) and Sinhala-English (Si-En), which are all sampled from Wikipedia, except for the Ru-En pair, which also contains sentences from Reddit. Additionally, the language-pairs used for development and testing also originate from Wikipedia: English-Czech (En-Cs), English-Japanese (En-Ja), Khmer-English (Km-En) and Pashto-English (Ps-En).

Finally, the new English-Marathi (En-Mr) data that is made available for train, development and testing this year is sampled from a combination of sources. More specifically the source side segments of the English-Marathi data contain segments from three different domains – healthcare, cultural, and general/news. The general domain and cultural domain data were obtained from the English (source side) segments in the IITB English-Hindi Parallel Corpus (Kunchukuttan et al., 2018). However, the healthcare domain data was obtained from publicly available NHS monolingual corpus[3].

All of the data was translated using large transformer-based NMT models, with established high performance for the languages in question. Specifically, for the language pairs in the training data (En-De, En-Zh, Et-En, Ne-En, Ru-En, Ro-En, Si-En), all source sentences were translated by a *fairseq* Transformer (Ott et al., 2019) bilingual model. The exception is the English-Marathi which was translated by the multilingual IndicTrans (En-X) Transformer-based NMT model, which was trained on the Samanantar parallel corpus (Ramesh et al., 2022).

For the languages provided in the development and test set, namely: En-Cz, En-Ja, Km-En and Ps-En we maintain the same we use the MBART50 (Tang et al., 2020),[4] to translate the source sentence of the other languages pairs, since it has been found to perform well, especially for low-resource languages (Tang et al., 2020). The En-Mr portion of the development and test data is translated similarly to the training data for this language pair.

Zero-shot language pair: This year we introduced a "surprise" language-pair, English-Yoruba (En-Yo), which represents a low-resource language pair. The Yoruba language is the third most spoken language in Africa, and it is native to southwestern Nigeria and the Republic of Benin (Eberhard et al., 2020). We extracted 1010 sentences in English from Wikipedia across 7 topics and translated them to Yoruba using Google Translate. Using adjusted guidelines from Fomicheva et al. (2021), we trained annotators to indicate sentence-level DA scores and to highlight erroneous words as word-level explanations for the DA scores.[5] On the 1010 sentences, they obtained agreements of 0.487 Pearson on sentence-level and 0.380 kappa on word-level. Note that in order to further encourage multilingual and unsupervised approaches, the setup for this zero-shot approach was slightly different to the previous edition, since we did not reveal the language pair before the release of the test data, and the zero-shot pair was included only in the multilingual sub-tasks for quality estimation

[3] The NHS corpus source sentences were crawled from the health directory of NHS available here: https://www.nhs.uk/conditions/

[4] https://github.com/pytorch/fairseq/tree/master/examples/multilingual

[5] Annotators were graduate students and native speakers of Yoruba and fluent in English.

(as opposed to a standalone subtask for this language pair only).

MQM data: As training data, we used annotations released for the Metrics shared task namely, the concatenation of the annotations released from Freitag et al. (2021a) with the annotations from last year Metrics task (Freitag et al., 2021b). Together, these annotations, cover 3 high-resource language pairs, namely: Chinese-English (Zh-En), English-German (En-De) and English-Russian (En-Ru), and span across two domains (News and Ted Talks). In contrast to DA, instead of one translation for each source, we have multiple translations coming from system participation's in the 2020 and 2021 News translation tasks (Barrault et al., 2020; Akhbardeh et al., 2021). For development set however, we follow an approach that is similar to the one use for the DA data: we translated the Newstest 2019 using a single NMT system, namely MBART50. Subsequently, for each language pair we asked an expert translator to provide MQM annotations. The test set was created similarly to the development, but instead of using Newstest 2019 we used the Newstest 2022 (the News data from this year's General MT shared task).

Overall, the released data for Task 1covers a total of 9 language pairs for training, 4 language pairs for development and 6 language pairs for testing including 1 zero-shot language pair. Statistics and details for each language pair are provided in Table 1.

2.1.1 Sentence-level quality prediction

There were two competition instances for the sentence-level sub-task. The first one focuses on DA- and the second one on MQM-derived annotations, both including a separate multilingual track. In the future, we aim to consolidate the competition instances into a single one for sentence-level, using our findings from this edition to align the annotation schemes in a better manner. We provide below the details for each annotation scheme and a comprehensive table with statistics for all annotations (Table 1).

DA annotations: For DA annotations, we followed the annotation and scoring conventions of previous editions. We provided MLQE-PE data (Fomicheva et al., 2022) used in previous years for training, which includes seven language pairs with $\approx$ 8,000 segments each. We also provided 26,000 segments of En-Mr which were annotated using the same annotation conventions. All translations were manually annotated for perceived quality, with a quality label ranging from 0 to 100, following the FLORES guidelines (Guzmán et al., 2019). According to the guidelines given to annotators, the 0-10 range represents an incorrect translation; 11-29, a translation with few correct keywords, but the overall meaning is different from the source; 30-50, a translation with major mistakes; 51-69, a translation which is understandable and conveys the overall meaning of the source but contains typos or grammatical errors; 70-90, a translation that closely preserves the semantics of the source sentence; and 91-100, a perfect translation. For each segment, there were at least three scores from independent raters (four in the case of En-Mr). DA scores were standardised using the z-score by rater, and the z-scores were provided as training targets. Participating systems are required to score sentences according to z-standardised DA scores.

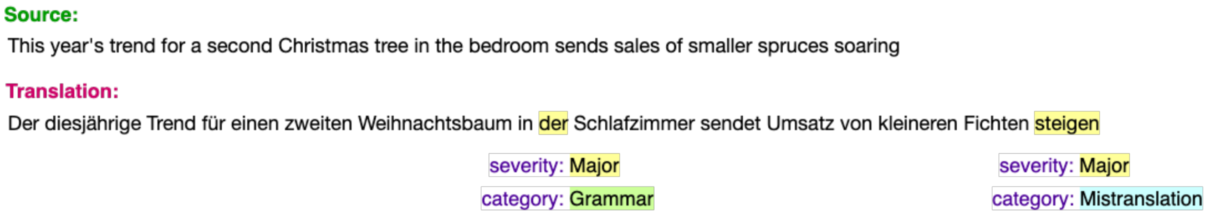
MQM annotations: As we have seen (§2.1), for the MQM annotations, we built on the available Google MQM annotations (Freitag et al., 2021a) that contain annotated data for the En-De and Zh-En data of WMT 2020 News Translation Systems (Barrault et al., 2020) as well as En-De, Zh-En and En-Ru annotations from WMT Metrics 2021 (Freitag et al., 2021b). These annotations, provided as training data, amount to more than 30,000 segments in total (see Table 1 for details per language pair). In addition, we provide newly annotated development and test sets for all three language pairs (En-De, En-Ru, Zh-En), amounting to approximately 1,000 segments per language pair.

Originally, MQM annotated segments include annotated erroneous text-spans on the translation side that are assigned two types of labels: (a) an error severity label {`minor`, `major`, `critical`} and (b) an error category label such as {`grammar`, `style/awkward`, `omission`, `mistranslation`}, `...`}. Each error severity is associated with a specific weight; hence a sentence score can be calculated for each segment based on these error weights. We demonstrate an example of MQM annotations and scores in Figure 1.

MQM scores according to Google weight scheme have the opposite direction of the DA scores since larger MQM scores denote worse translation quality, i.e., a larger number of errors or more severe errors. To address this inconsistency, we

Language Pairs	Sentences Train / Dev / Test22	Tokens Train / Dev / Test22	DA	PE	MQM	CE	Data Source
En-De [1]	9,000 / 1,000 / –	131,499 / 16,545 / –	✓	✓			Wikipedia
En-Zh	9,000 / 1,000 / –	131,892 / 16,637 / –	✓	✓			Wikipedia
Ru-En	9,000 / 1,000 / –	94,221 / 11,650 / –	✓	✓			Reddit
Ro-En	9,000 / 1,000 / –	137,466 / 17,359 / –	✓	✓			Wikipedia
Et-En	9,000 / 1,000 / –	112,503 / 14,044 / –	✓	✓			Wikipedia
Ne-En	9,000 / 1,000 / –	120,078 / 15,017 / –	✓	✓			Wikipedia
Si-En	9,000 / 1,000 / –	125,223 / 15,709 / –	✓	✓			Wikipedia
En-Mr	26,000 / 1,000 / 1,000	690,532 / 27,049 / 26,253	✓	✓			
Ps-En	– / 1,000 / 1,000	– / 27,045 / 27,414	✓	✓			Wikipedia
Km-En	– / 1,000 / 1,000	– / 21,981 / 22,048	✓	✓			Wikipedia
En-Ja	– / 1,000 / 1,000	– / 20,626 / 20,646	✓	✓			Wikipedia
En-Cs	– / 1,000 / 1,000	– / 20,394 /20,244	✓	✓			Wikipedia
En-Yo	– / – / 1,010	– / – / 21,238	✓	✓			
En-De [2]	28,909 / 1,005 / 511	839,473 / 24,373 / 13,220			✓		WMT-newstest
En-Ru	15,628 / 1,005 / 511	357,452 / 24,373 / 13,220			✓		WMT-newstest
Zh-En	35,327 / 1,019 / 505	1,586,883 / 51,969 / 15,602			✓		WMT-newstest
En-De	155,511 / 17,280 / 500	8,193,693 / 915,061 / 27,771				✓	News-Commentary
Pt-En	39,926 / 4,437 / 500	2,281,515 / 253,594 / 29,794				✓	News-Commentary

Table 1: Statistics of the data used for Task 1 (DA), Task 2 (PE) and Task 3 (CE) (last four rows). The number of tokens is computed based on the source sentences.



Source:

This year's trend for a second Christmas tree in the bedroom sends sales of smaller spruces soaring

Translation:

Der diesjährige Trend für einen zweiten Weihnachtsbaum in der Schlafzimmer sendet Umsatz von kleineren Fichten steigen

severity: Major
category: Grammar

severity: Major
category: Mistranslation

Figure 1: Example of MQM annotations on the target (translation) side, on a English–German (En-De) sentence pair.

invert the MQM scores and standardise per annotator. For training data we had access to multiple annotations per segment and calculated an average score after standardisation, keeping also the original MQM scores per annotator, to allow the participants to take full advantage of the different annotations (Basile et al., 2021). For the same reasons, we opted not to aggregate the annotated text-spans.

Regarding evaluation, systems in this task (both for DA and MQM) are **evaluated against the true z-normalised sentence scores using Spearman's rank correlation coefficient** ρ **as the primary metric**. This is what was used for ranking system submissions. Pearson's correlation coefficient, r, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) were also computed as secondary metrics but not used for the final ranking between systems.

2.1.2 Word-level quality prediction

This sub-task focuses on detecting word-level errors in the MT output. The goal is to automatically predict the quality of each token using a binary decision, i.e., using OK as a label for tokens translated correctly and BAD otherwise. We deviate from the annotation pattern of previous years in that, we do not consider annotations of the gaps between tokens or source-side annotations. Instead, to account for omission errors, we consider the following convention: the token on the right side of the omitted text in the translation is annotated as "BAD". An additional `<EOS>` token is appended at the end of every translation segment to account for omissions at the end of each sentence. This allows the provision of a unified framework for both the post-edit originated annotations and the MQM annotations.

We thus use the same source-translation pairs used for the sentence-level tasks and obtain the binary tags as follows:

- For post-edited data, we use TER (Snover et al., 2006) to obtain alignments between translation and post-edit and annotate the misaligned tokens as BAD.

- For MQM data, the tokens that fall within the text-spans annotated as errors (or any severity or category) are annotated as BAD. If the whitespace between two words is annotated as an error, then this is considered an omission, and the next token is annotated as BAD.

Participants were encouraged to submit for each language pair and also for the **multilingual variants** of each sub-task. For the DA-based sentence-level competition, as well as the word-level sub-task, there was an additional multilingual variant that included the zero-shot language pair (En-Yo). The latter aimed at fostering work on language-independent models, as well as models that are truly multilingual.

For word-level task, **submissions are ranked using the Matthews correlation coefficient (MCC) as the primary metric**, while F1-scores are provided as complementary information.

2.2 Task 2: Explainable Quality Estimation

Following the success of the shared task on Explainable Quality Estimation organized by the Eval4NLP workshop in 2021 (Fomicheva et al., 2021), in this sub-task we aim to address translation error identification as rationale extraction from sentence-level quality estimation systems. If a QE system reasonably estimates the quality of a translated sentence, an explanation extracted from the system should indicate word-level translation errors in the input (if any) as reasons for imperfect sentence-level scores. Particularly, for each input pair of source and target sentences, participating teams are asked to provide (*i*) a sentence-level score estimating the translation quality and (*ii*) a list of continuous word-level scores where the tokens with the highest scores are expected to correspond to translation errors considered relevant by human annotators.

In this explainable QE task, we use all the nine language pairs and their word-level test sets from Task 1 (see §2.1.2) with En-Yo being a separate language pair (rather than blending it in the multilingual test set). Therefore, the participants are allowed to use the sentence-level scores from the datasets in Task 1 to train their sentence-level models in Task 2. However, as Task 2 aims to promote the research on the explainability of QE systems, we encourage the participants to use or develop explanation methods to identify contributions of words or tokens in the input. Unlike Task 1, **the participants of Task 2 are not allowed to supervise their models with any token-level or word-level labels or signals (whether they are from natural or synthetic data) in order to directly predict word-level errors.** Consequently, we do not require the participants to convert their word-level scores into predicted binary labels (OK/BAD) since this process usually requires a word-level QE dataset to search for an optimum score threshold.

Concerning the evaluation of this task, we focus on assessing the quality of explanations (i.e., the submitted word-level scores), not the sentence-level predictions. Specifically, we measure how well the word-level scores provided by the participants correspond with human word-level error annotations, which are binary ground truth labels. Unlike the Eval4NLP 2021 shared task, which ranked participating systems by a combination of three metrics (Fomicheva et al., 2021), **we use *Recall at Top-K*, also known as *R-precision* in information retrieval literature (Manning et al., 2008, chapter 8), as the primary metric this year** due to two reasons. First, it is preferable to have a single main metric to avoid confusion and also some potential side effects that combining the three metrics might produce. Second, Recall at Top-K seemed to help discriminate best between the participating submissions in the Eval4NLP shared task. Assume that, for a given pair of source and target sentences, there are K words annotated as translation errors by humans. Recall at Top-K equals $\frac{r}{K}$ when there are r out of the K error words appearing in the list of top-K words ranked by the submitted word-level scores descendingly. In addition, AUC (an area under the receiver operating characteristic curve) and AP (average precision) are used as secondary metrics. Considering the word level, AUC summarises the curve between true positive rate and false positive rate, while AP summarises the curve between precision and recall. For both of the secondary metrics, higher values are the better. Although we report metrics for sentence-level predictions, including Pearson's correlation and Spearman's correlation, as additional information, we do not use them for ranking the participants or determining the winner in this explainability task.

2.3 Task 3: Critical Error Detection

In this sub-task, we reshape the binary classification task introduced in last year's edition (Specia et al., 2021) to predict whether the translated sentence contains (at least) one critical error.

Following Specia et al. (2021), we consider that a translation contains a critical error if it deviates from the meaning of the source sentence in such a way that it is misleading and may lead to several implications. As noted by Specia et al. (2021), deviations in meaning can happen in three ways: mistranslation errors have critical content translated incorrectly into a different meaning; hallucination errors introduce critical content in the translation that is not in the source; and deletion errors remove critical content that is in the source from the translation.

In this task, we focus on five critical error categories:

- Additions: The content of the translation is only partially supported by the source.
- Deletions: Part of the source sentence is ignored by the MT engine.
- Named Entities: A named entity (people, organization, location, etc.) is mistranslated into another incorrect named entity.
- Meaning: The translated sentence either introduces or removes a negation and the sentence meaning is completely reversed.
- Numbers: The MT system translates a number/date/time or unit incorrectly.

For this task, we introduce a new dataset obtained by perturbing a corpus of News articles with SMAUG (Alves et al., 2022) and using humans to validate perturbation on the test set. The original data for this task is composed of the News articles from OPUS News-Commentary (Tiedemann, 2012) for the language pairs English-German and Portuguese-English.

For the English-German language pairs, there are no Deviation in Meaning errors, as the perturbation is only available for into English language pairs. The new dataset is purposefully unbalanced, as these phenomena are rare, containing approximately 5% of translations with critical errors. Table 1 presents the number of records for each language pair.

Since the dataset for this task is artificially generated, the participants were encouraged to submit systems that did not rely on the provided training data. As such, submissions were split into two groups: *unconstrained* and *constrained*. In the first group, the participants have access to the training data. In the second, the systems should only be trained on quality scores such as DA, HTER and MQM annotations. With this setting, we aim to evaluate whether systems can identify critical errors while maintaining correlations with human judgements.

In the evaluation of this task, the participants were not required to submit any classification threshold for their systems. For the *unconstrained* setting, the systems are specifically trained to detect errors and should output high scores for translations containing these errors. As such, for each language-pair, we considered as positive predictions the K records with highest scores, where K is the number of positive records for that language-pair in the test set. Regarding the *constrained* setting, these systems are only trained on quality scores and are expected to assign lower scores to translations with critical errors. Therefore, we considered the K records with lowest scores as positive predictions. From here, we measured the MCC, *Recall* and *Precision* for each submission.

3 Baseline systems

Task 1: Quality Estimation baseline systems: For Task 1, both for word and sentence-level, we used a multilingual transformer-based Predictor-Estimator approach (Kim et al., 2017), which is described in detail in Fomicheva et al. (2022). For the implementation and training we use the OpenKiwi (Kepler et al., 2019) framework. We trained the baseline model using a multilingual and multitask setting and training jointly on the sentence-level scores and word-level tags. For the word-level loss, $\mathcal{L}_{\text{word}}$, the weight of BAD tags is multiplied by a factor of $\lambda_{BAD} = 3.0$, but the sentence- and word-level loss have equal weight in the overall joint loss estimation: $\mathcal{L} = \mathcal{L}_{\text{word}} + \mathcal{L}_{\text{sent}}$. We trained different baselines for the DA/post-edit originated language pairs and the MQM originated language-pairs.

For the DA/post-edit baseline, the model was trained using the DA scores as sentence targets and the OK/BAD tags as word targets. For training we used the concatenated data for all language pairs available under training data and used the concatenation of the additional language pairs that were made available in the development set as validation. We trained two baselines with this setup, using different encoders for the encoding (predictor) part of the architecture: (a) XLM-R transformer with the `xlm-roberta-large` model and (b) RemBERT model which has been pre-trained on additional languages that include Yoruba and can hence account for the zero-shot language.

For the MQM baseline, the model was trained using the normalised and inverted MQM scores as sentence targets and the OK/BAD tags as word targets. The baseline model was trained using the concatenated training data for all three language pairs and used the concatenated development data for the same pairs as the validation set. The XLM-R transformer with the `xlm-roberta-large` model was used as an encoder.

Task 2: Explainability baseline systems: We provide two baseline systems for Task 2. One is a random baseline where we sampled scores uniformly at random from a continuous [0..1) range for each target token and for a sentence-level score. The other one is a combination of a supervised quality annotation model, OpenKiwi (Kepler et al., 2019) and LIME (Ribeiro et al., 2016) where OpenKiwi is used to predict sentence-level quality scores while LIME is used to compute, for every token in the target sentence, its importance for the sentence-level quality score returned by OpenKiwi. For the OpenKiwi implementation we used a similar setup described for the baselines of Task 1, but we trained the OpenKiwi model using only sentence-level supervision, to align with the task requirements. We trained two multilingual instances, one on DA- and one on MQM-derived data, using XLM-R large encoder in both cases.

LIME is a model-agnostic post-hoc explanation method which trains a linear model to estimate the behavior of a target model (i.e., OpenKiwi in our case) around an input example to be explained so the weights of the linear model correspond to the importance of individual input tokens. Because higher sentence-level scores in our gold standard mean better translation quality, we invert token-level scores generated by LIME so that higher values correspond to errors as required by the task description.

Task 3: Critical Error Detection baseline systems: For task 3, we consider a baseline system for each setting.

In the *constrained* setting, we considered COMET-QE (Rei et al., 2021)[6], which was a top-performing QE-as-a-Metric system in last years Metrics shared task (Freitag et al., 2021b).

Regarding the *unconstrained* setting, we fine-tune an `xlm-roberta-large` model using the COMET framework (Rei et al., 2020). Both the source and translation are jointly encoded into a vector representation which is the input of a final estimator that predicts the probability of the translation containing a critical error. Here, the estimator weights are randomly initialised. We fine-tune the model on the provided training data for a maximum of 5 epochs. At the end of each epoch, we perform a validation step by measuring the MCC on the validation set considering a classification threshold of 0.5. We select the model with the highest MCC on the validation data.

4 Participants

Alibaba-Translate (T1-DA): For the DA subtask, the team participated in all language pairs except the zero-shot LP. The implemented system (Wang et al., 2021), uses glass-box QE features to estimate the uncertainty of machine translation segments and incorporates the features into the transfer learning from the large-scale pre-trained model, XLM-R. The participants used exclusively the DA data provided for this edition of the QE shared task. Of the provided data, the 7 language pairs except for English-Marathi, were combined to train a multilingual model. For English-Marathi, a separate bilingual model was trained. For the final submission the participants ensembled multiple checkpoints.

(T1-MQM): The submission for sentence-level MQM task is based on a multilingual unified framework for translation evaluation. The applied framework UniTE (Wan et al., 2022) considers three input formats – source-only (QE or reference-free metric), reference-only and source-reference-combined. The participants used synthetic datasets with pseudo labels during continuous pre-training phase, and fine-tuned with DA and MQM training

[6] More precisely we used the `wmt21-comet-qe-mqm` model

datasets from the year 2017 to 2021. To obtain the final model predictions they use the source-only evaluation. For multilingual phase, they ensembled predictions using two different backbones – one using XLM-R encoder and the other using InfoXLM. For the ensembling, they picked the best 2 checkpoints on the development dataset.

BJTU-Toshiba (T1-MQM): BJTU-Toshiba participation focused on ensembling different models and using external data. They ensemble multiple pre-trained models, both monolingual and bilingual. The monolingual models are trained only on the text of the target language. Specifically, they use monolingual BERT, Roberta, and Electra-discriminator as the monolingual extractor, and XLM-R as the bilingual extractor. They also use in-domain parallel data to fine-tune and adapt the pre-trained models to the target language and domain. The in-domain data is selected by a BERT-classifier from the parallel data provided by the news translation task, and for each direction, they end up using roughly 1 million sentence pairs for fine-tuning. They explore two styles of fine-tuning, namely Translation Language Model and Replaced Token Detection. For Replaced Token Detection, they use the first 1/3 layers of the model as generator, and after the training they drop the generator and only use the discriminator as the feature extractor.

HW-TSC (T1): HW-TSC's submission follows Predictor-Estimator framework with a pre-trained XLM-R Predictor, a feed-forward Estimator for sentence-level QE subtask and a binary classifier Estimator for word-level QE subtask. Specially, the Predictor is a cross-lingual language model that receives source and target tokens concatenated and returns representations that attend to both languages. WMT 2022's news translation task training data is been used to train the Predictor using a cross-lingual masked language model objective. All of the WMT QE 2022 DA and MQM training data are used to train two different multilingual QE models, one for sentence-level and another one for word-level.

(T2:) The language encoder trained for Task 1 is being used to get source and target token embeddings. After computing cosine similarity between target and source token embeddings, the max cosine similarity of each target token to all the source tokens is selected as quality score. Intuitively, a low score means the target token is more likely to be an error (lack of good alignment), so every target word quality score is multiplied by a negative value.

HyperMT - aiXplain (T1-all): The system is trained with AutoML functionalities in FLAML framework using lightgbm estimator. It utilizes COMET-QE score as feature along-side with many other linguistic features extracted with Stanza from source texts and their translations: the number of tokens, characters, and the average word length of sentences; the frequency of Part-of-Speech and Named Entity Recognition labels, and the frequency of morphological features. The differences in values of linguistic features between source texts and translations are also included as features. This allows the system to work in multilingual settings as well.

IST-Unbabel (T1-all): IST-Unbabel team proposed an extension of COMET, dubbed COMET-Kiwi, which includes a word-level layer and can be trained on both sentence-level scores and word-level labels in a multi-tasking fashion. Their final submission for task 1 is a weighted ensemble between models trained using InfoXLM (Chi et al., 2021) and RemBERT (Chung et al., 2021). All these models are pretrained on the data from the metrics shared tasks and, for word-level, they pretrained on both QT21 and APE-Quest datasets.

(T2) For the second task they use the COMET-Kiwi framework as the backbone of a sentence-level QE model and added layer and headwise parameters to the QE model: for each layer and for each head, they train individual parameters to construct a sparse distribution over the layers/heads to better leverage these representations. They leveraged different encoders – InfoXLM and RemBERT – and used them individually as the backbone of our QE sentence-level models. The models used to extract explanations were multilingual ones trained for DA and MQM separately. The explainability weights were obtained from the attention weights scaled by the norm of the gradient of the value vectors (Chrysostomou and Aletras, 2022). No word supervision was used and all explanations were extracted relying solely on models that produced the sentence-level scores. The final submissions are ensembles of explanations from different attention layers/heads according to the validation data. For the zero-shot language pair (En-Yo), they created an ensemble with the attention layers/heads that were among the top-performing ensembles for other language pairs.

(T3) For task 3 a single model from task 1 using InfoXLM encoder and trained on DA annotations was submitted.

KU X Upstage (T3): KU X Upstage employs an XLM-R large model without leveraging any additional parallel corpus. Instead, they attempt to maximise its capability by adopting prompt-based fine-tuning, which reformulates the Critical Error Detection task as a masked language modelling objective (a pre-training strategy of this model) before training. They generate hard prompts suitable for QE task through prompt engineering, and templates consist largely of three types according to the information utilised: naive template, template with a contrastive demo, and template with Google Translate. The final score is obtained by extracting the probability of a word mapped to BAD among verbalizers. They gain an additional performance boost from the template ensemble by adding the values from multiple templates.

NJUNLP (T1-all): NJUNLP submission makes use of pseudo data and multi-task learning. Inspired by DirectQE (Cui et al., 2021), they experiment with several novel methods to generate pseudo data for all three subtasks (MQM, DA, and PE) using the conditional masked language model and the NMT model to generate high quality synthetic data and pseudo labels. The proposed methods control the decoding process to generate more fluent pseudo translations close to the actual distribution of the gold data. They pre-train the XLM-R large model with the generated pseudo data and then fine-tune this model with the real QE task data, using multi-task learning in both stages. They jointly learn sentence-level scores (with regression and rank tasks) and word-level tags (with a sequence tagging task). For the final submissions they ensemble sentence-level results by averaging all valid output scores and ensemble word-level results using a voting mechanism. For the pseudo label generation they use publicly available parallel data, specifically: the data provided by the WMT translation task for En-De (9M), En-Ru (3M), and Zh-En (3M) language pairs. The 660K parallel sentences from OPUS[7] for the Km-En language pair. They also use 3.6M parallel data from the target translation model[8] for the En-Mr language pair, as well as WMT2017, WMT2019, and WMT2020 En-De PE data for the En-De language pair.

Papago (T1-full): Papago submitted a multilingual and multi-task model, trained to predict jointly both sentence and word level. The system's architecture consists of Pretrained Language Model with task independent layers optimized for both sentence and word level quality prediction. They propose an auxiliary loss function to the final objective function to further improve performance. They also augment training data by either generating (i.e. pseudo data) or collecting open source data that is deemed to be relevant to QE task. Finally, they train and select the checkpoints for the final submission with cross-validation for better generalization and ensemble multiple models for their final submission.

UCBerkeley-UMD (T1:DA): UCBerkeley-UMD used a large-scale multilingual model to back translate from Czech to English. They compared the quality of the Czech translation by examining the translation from Czech back to English with the original source text in English. This is motivated by literature that humans tend to perform quality checks on translations when they do not understand the target language.

UT-QE (T2): The UT-QE team used XLMR-Score (Azadi et al., 2022) as an unsupervised sentence-level metric, which is computed as BERTScore but in a cross-lingual manner while using the XLM-R model. The matched

[7] https://opus.nlpl.eu/

[8] https://indicnlp.ai4bharat.org/indic-trans/

ID	Affiliations	
Alibaba Translate	DAMO Academy, Alibaba Group & University of Science and Technology of China & CT Lab, University of Macau, China & National University of Singapore, Republic of Singapore	(Bao et al., 2022)
BJTU-Toshiba	Beijing Jiaotong University, China & Toshiba Co., Ltd.	(Huang et al., 2022)
HW-TSC	Huawei Translation Services Center & Nanjing University, China	(Su et al., 2022)
HyperMT - aiXplain	aiXplain	–
IST-Unbabel	INESC-ID & Instituto de Telecomunicações & Instituto Superior Técnico & Unbabel, Portugal	(Rei et al., 2022)
KU X Upstage	Korea University, Korea & Upstage	(Eo et al., 2022)
NJUNLP	Huawei Translation Services Center, China	(Geng et al., 2022)
Papago	Papago, Naver Corp	(Lim and Park, 2022)
UCBerkeley-UMD	University of California, Berkeley & University of Maryland	(Mehandru et al., 2022)
UT-QE	University of Tehran, Iran	(Azadi et al., 2022)
Welocalize-ARC/NKUA	Welocalize Inc, USA & National Kapodistrian University & Athena RC, Greece	(Zafeiridou and Sofianopoulos, 2022)

Table 2: Participants to the WMT22 Quality Estimation shared task.

tokens distances in this metric were used as token-level scores. In order to alleviate the mismatching issues, they also try to fine-tune the XLM-R model on word alignments from parallel corpora to make it represent the aligned words in different languages closer to each other, and use the fine-tuned model instead of XLM-R for scoring sentences and tokens.

Welocalize-ARC/NKUA (T1-DA): Welocalize-ARC/NKUA's submission for the Task 1 follows the Predictor-Estimator framework (Kim et al., 2017) with a regression head on top to estimate the z-standardised DA. More specifically, they use a pre-trained Transformer for feature extraction and then concatenate the extracted features with additional glass-box features. The glass-box features are also produced using pre-trained models and by applying multiple techniques to estimate different types of uncertainty for each translated sentence. The final features are then used as input for the QE regression model, which is a simple sequential Neural Network with a linear output layer. Finally, the performance of the model is optimised by employing Monte Carlo Dropout during both training and inference. Regarding the data, they use only the provided datasets (the MLQE-PE train/dev sets along with the additional dataset for Marathi language) as well as some of the provided additional training resources of the Metrics shared task.

Table 2 lists all participating teams submitting systems to any of the tasks, and Table 3 report the number of successful submissions to each of the sub-tasks and language pairs. Each team was allowed up to ten submissions for each task variant and language pair (with a limit of two submissions per day). In the descriptions below, participation in specific tasks is denoted by a task identifier (T1 = Task 1, T2 = Task 2, T3 = Task 3).

5 Results

In this section, we present and discuss the results of our shared task. Please note that for all the three subtasks we used statistical significance testing with $p = 0.05$.

5.1 Task 1

As we have seen in Task 1 description (§2.1.1), submissions are evaluated against the true z-normalised sentence scores using Spearman's rank correlation coefficient ρ along with the following secondary metrics: Pearson's correlation coefficient, r, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). Nonetheless, **the final ranking between systems is calculated using the primary metric only (Spearman's ρ).** Also, statistical significance was computed using William's test.[9]

For the Task 1 word-level task, the submissions are ranked using the Matthews correlation coeffi-

[9] https://github.com/ygraham/mt-qe-eval

Task/LP	# submission
Task 1 – Sent-level Direct Assessment	**161**
Multilingual w/o En-Yo	21
Multilingual w En-Yo	23
English-Marathi	24
English-Czech	33
English-Japanese	22
Pashto-English	16
Khmer-English	22
Task 1 – Sent-level MQM	**402**
Multilingual	38
English-German	65
English-Russian	62
Chinese-English	76
Task 1 – Word-level	**247**
Multilingual w/o En-Yo	18
Multilingual w En-Yo	17
English-Czech	32
English-Japanese	27
English-Marathi	24
Pashto-English	13
Khmer-English	28
English-German	28
English-Russian	18
Chinese-English	27
Task 2 – Explainable QE	**161**
English-Czech	14
English-Japanese	14
English-Marathi	13
Pashto-English	30
Khmer-English	25
English-German	17
English-Russian	12
Chinese-English	12
English-Yoruba	12
Task 3 – Sent-Level Critical Error Det.	**20**
Constrained	
English-German	2
Portuguese-English	2
Unconstrained	
English-German	10
Portuguese-English	6
Total	**991**

Table 3: Number of submissions to each sub-task and language-pair at the WMT22 Quality Estimation shared task.

cient (MCC). F1-scores are provided as complementary information only and statistical significance was computed using randomisation tests (Yeh, 2000) with Bonferroni correction (Abdi, 2007) for each language pair.

The majority of participants implemented multilingual models and the top performing systems adopted a multi-tasking approach, learning the sentence- and word-level targets jointly (IST-Unbabel, Papago, NJUNLP). It is important to note that all participants relied on large pre-trained encoders (XLM-R, RemBERT, BERT, ELECTRA), which seems to be the norm for high-performance in quality estimation, but can constitute a limitation for performance in truly multi-lingual scenarios where the target languages are not seen during pre-training. Additionally, many final submissions consisted of ensembles combining different large pretrained models increasing even further the total number of model parameters.

Another trend that seems to carry on from previous editions of the task is the incorporation of additional features in QE models (glass-box features were incorporated in Alibaba's DA systems while linguistic features were incorporated in aiXplain QE system), however in this edition such approaches were outperformed by models that put more emphasis on pre-training, using auxiliary tasks and external data.

For the sentence-level sub-tasks, participants managed to achieve high correlations for the majority of language pairs, especially for the DA originated data, with the exception of En-Ja. The results show an improvement compared to the last edition, although it is hard to draw a direct comparison due to changes in the available train/development data. However, it is interesting to note that performance for En-Mr, for which we provided considerable more data than for the other language pairs is still in the same range as results for the other language pairs. It would thus be interesting to investigate further which properties render a language pair harder to evaluate.

For the MQM data the overall correlations achieved were lower in comparison to the DA ones although still meaningful. Note that compared to the DA data, the MQM language pairs were high-resource ones, which could also influence performance. Additionally, small discrepancies between the annotation guidelines in the train set and the dev/test sets could have further complicated the task. We intend to further investigate the MQM potential in future editions, with the addition of new language pairs and more annotated data.

For the word-level subtask, IST-Unbabel, NJUNLP and Papago tied at the top for most language pairs, and we can observe that correlations are moderate across language pairs (both DA and MQM originated ones). It is important to note that no team seems to have submitted predictions using a word-level only supervision; instead all the participants of this task used a multi-task approach, learning jointly word and sentence level scores.

Model	Multi	Multi (w/o En-Yo)	En-Cs	En-Ja	En-Mr	Km-En	Ps-En
IST-Unbabel	**0.572**	**0.605**	**0.655**	**0.385**	**0.592**	**0.669**	**0.722**
Papago	0.502	0.571	**0.636**	0.327	**0.604**	0.653	0.671
Alibaba Translate	–	0.585	0.635	0.348	**0.597**	0.657	0.697
Welocalize-ARC/NKUA	0.448	0.506	0.563	0.276	0.444	0.623	–
BASELINE	0.415	0.497	0.560	0.272	0.436	0.579	0.641
lp_sunny‡	0.414	0.485	0.511	0.290	0.395	0.611	0.637
HW-TSC	–	–	0.626	0.341	0.567	0.509	0.661
aiXplain	–	–	0.477	0.274	0.493	–	–
NJUNLP	–	–	–	–	**0.585**	–	–
UCBerkeley-UMD*	–	–	0.285	–	–	–	–

Table 4: Spearman correlation with **Direct Assessments** for the submissions to WMT22 Quality Estimation **Task 1**. For each language pair, results marked in bold correspond to the winning submissions, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey; ‡ indicates Codalab username of participants from whom we have not received further information and * indicates late submissions that were not considered for the official ranking of participating systems

Model	Multi	En-De	En-Ru	Zh-En
IST-Unbabel	**0.474**	0.561	**0.519**	**0.348**
NJUNLP	0.468	**0.635**	**0.474**	**0.296**
Alibaba-Translate	0.456	0.550	**0.505**	**0.347**
Papago	0.449	0.582	**0.496**	**0.325**
lp_sunny ‡	0.415	0.495	0.453	0.298
BASELINE	0.317	0.455	0.333	0.164
BJTU-Toshiba	–	**0.621**	0.434	**0.299**
HW-TSC	–	0.494	0.433	0.369
aiXplain	–	0.376	0.338	0.194
pu_nlp ‡	–	0.611	–	–

Table 5: Spearman correlation with **MQM** for the submissions to WMT22 Quality Estimation **Task 1**. For each language pair, results marked in bold correspond to the winning submissions, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey; ‡ indicates Codalab username of participants from whom we have not received further information.

Best performers The scores in Tables 4 - 6 show the participant scores for the main metric, ordered by the best performance in the multilingual subtasks. IST-Unbabel is the clear winner for the multilingual subtasks, but for the individual language pairs results vary and multiple participants are tied at the top. All top-performing approaches (IST-Unbabel, Papago, NJUNLP and Alibaba) share some common characteristics: (1) they constitute multilingual and multi-task approaches; (2) they use external data during pre-training, either adapted from other tasks (such as the Metrics task (Freitag et al., 2022)) or generated artificially (pseudo data); and (3) they use ensembling for the final submission.

5.2 Task 2

Three teams participated in Task 2, IST-Unbabel, HW-TSC and UT-QE. IST-Unbabel participated in all 9 language pairs, HW-TSC in all languages pairs except English-Yoruba, and UT-QE only in Khmer-English and Pashto-English. As shown in Table 7, IST-Unbabel wins 7 of 9 LPs according to the metric Recall at Top-K, HW-TSC the remaining 2. With Bonferroni correction, IST-Unbabel wins 4 LPs, HW-TSC wins 2, and both are indistinguishable on the remaining 3 LPs. Average precision (AP) yields identical results as Recall at Top-K in terms of ranking of the teams. There is one difference according to the metric AUC in terms of winners: HW-TSC wins English-Japanese. Finally, all participating teams beat both baselines in all cases.

For sentence-level performance (see Appendix D), IST-Unbabel wins all LPs according to Pearson's correlation and all LPs according to Spearman's correlation except for Khmer-English, which HW-TSC wins. Not all teams beat all baselines in terms of sentence-level performance.

The winning teams obtain the lowest sentence-level correlations for English-Chinese, English-Japanese and English-Yoruba and the highest correlations for Khmer-English and English-German. This may be related to the quality of annotations and the quality of MT systems involved. For word-level explainability scores, the lowest Recall at Top-K scores are obtained for English-Yoruba and English-Marathi, whereas the highest scores are obtained for Pashto-English and Khmer-English. The fact that the winning systems obtain low sentence and word-level scores for English-Yoruba and high scores for Khmer-English may indicate that the

Model	Multi	Multi (w/o En-Yo)	En-Cs	En-Ja	En-Mr	Kh-En	Ps-En	En-De	En-Ru	Zh-En
IST-Unbabel	**0.341**	**0.361**	**0.436**	**0.238**	**0.392**	**0.425**	**0.424**	**0.303**	**0.427**	**0.360**
Papago	0.317	**0.343**	**0.396**	**0.257**	**0.418**	**0.429**	0.374	**0.319**	**0.421**	**0.351**
BASELINE	0.235	0.257	0.325	0.175	0.306	0.402	0.359	0.182	0.203	0.104
HW-TSC	–	0.218	**0.424**	**0.258**	0.351	0.353	0.358	0.274	0.343	0.246
NJUNLP	–	–	–	–	**0.412**	**0.421**	–	**0.352**	**0.390**	**0.308**

Table 6: **Matthew Correlation Coefficient** (MCC) for the submissions to WMT22 Quality Estimation **Task 1 (word-level)**. For each language pair, results marked in bold correspond to the winning submissions, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Model	En-Cs	En-Ja	En-Mr	En-Ru	En-De	En-Yo	Km-En	Ps-En	Zh-En
IST-Unbabel	**0.561**	**0.466**	**0.317**	**0.390**	**0.365**	**0.234**	0.665	0.672	**0.379**
HW-TSC	**0.536**	**0.462**	**0.280**	0.313	0.252	–	**0.686**	**0.715**	0.220
BASELINE (OpenKiwi+LIME)	0.417	0.367	0.194	0.135	0.074	0.111	0.580	0.615	0.048
BASELINE (Random)	0.363	0.336	0.167	0.148	0.124	0.144	0.565	0.614	0.093
UT-QE	–	–	–	–	–	–	0.622	0.668	–

Table 7: **Recall at Top-K** for the submissions to the WMT22 Quality Estimation **Task 2 (Explainable QE)**. For each language pair, results marked in bold correspond to the winning submissions, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Model	En-De (Cons)	En-De (UN-cons)	Pt-En (Cons)	Pt-En (UN-cons)
KU X Upstage	–	0.964	–	0.984
IST-Unbabel	0.564	–	0.721	–
BASELINE	0.074	0.855	-0.001	0.934
aiXplain	–	0.219	–	0.179

Table 8: **Matthews Correlation Coefficient** (MCC) for the submissions to WMT21 Quality Estimation **Task 3 (Critical Error Detection)**. For each language pair, results marked in bold correspond to the winning submissions, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

tasks are correlated (as one may intuitively expect): a QE system that yields better sentence-level scores also highlights word-level errors more correctly.

5.3 Task 3

In this task, we divide participants into *unconstrained* and *constrained* settings, and address each group in separate. As in the last year, this task attracted few participants, which we attribute to the recentness of the task.

In the *unconstrained* setting, there are two participants: KU X Upstage and HyperMT - aiXplain. The first achieved very high values for the measured metrics, and is the best performer for this setting for both language pairs. The second obtained lower values, falling below the baseline on both language pairs.

In the *constrained* setting, a single submission was received: IST-Unbabel. Their system outperformed the baseline on both language pairs.

6 Discussion

In what follows, we discuss the main findings of this year's shared task based on the goals we had previously identified for it.

General progress Participating systems achieved very promising results for most languages, including the newly introduced language-pairs as well as the new annotation style (MQM). The **best performing submissions showed moderate to strong correlation for sentence-level DA and MQM prediction tasks.** While it is hard to draw direct comparisons with the previous editions, the overall correlation scores obtained are similar or improved for the common language-pairs. In combination with the outcomes of previous editions, it seems that multi-lingual and multi-task systems that are able to take advantage of multiple resources, are showing better and more robust results. However, the **word-level quality prediction is still a challenging task and there is ample room for improvement**. Along the same lines, further **exploring explainability tasks, that support the sentence level predictions with word level scores seems a promising path to motivate finer-grained approaches to word-level** quality annotations.

DA vs MQM annotations To further understand the observed discrepancies between top performances in the DA and MQM sub-tasks for sentence-level quality estimation, we analyse the distributions of predicted scores vs gold scores for each language pair, as presented in Figure 2.

We can see in the scatter plots that there are multiple test-segments which are annotated as perfect translations (maximum possible normalised MQM score), which fail to be classified accordingly as indicated by the top parts of the MQM scatter plots in Figure 2. Overall, even with DA annotations we can see that **language pairs with more balanced distribution between high and low quality segments (Km-En, Ps-En) are those for which QE systems obtain better correlations**, compare to more skewed language pairs (En-Mr, En-Ja).

Additionally, we can see that the **MQM scores are significantly skewed towards higher scores**, with long-tails of few very low quality instances. This provides motivation to revisit the quantification of MQM annotations to generate sentence level scores and further experiments into consolidating MQM annotations from different annotators. Furthermore, perhaps providing access to finer-grained MQM annotations (using the category or severity labels as targets) could aid in obtaining more meaningful outcomes. In future editions we intend to further expand the coverage of languages for MQM annotations that will allow us to draw further conclusions and push the state-of-the-art further in this track.

Zero shot predictions We found that **even without development data or prior knowledge about the language pair, the systems that submitted predictions for En-Yo still achieved meaningful correlations**. For the quality assessment and explainability tasks, the achieved correlations are lower compared to the "seen" language pairs, but still comparable. We can also observe the scatter plot distributions that show the correlation obtained by the top performing system that is comparable with the other DA distributions.

However, we noticed that the **availability of the zero-shot languages in the frequently used pretrained encoders posed an additional challenge** for the participants as the performance on En-Yo seemed dependent on whether the pretrained language model had seen Yoruba text during pretraining. In future editions, we hope that mixing different zero-shot languages will further motivate unsupervised approaches.

Explainable quality estimation The performance of the baselines in Task 2 suggests that applying a model-agnostic explanation method (i.e., LIME) to a relatively good sentence-level QE system (i.e., OpenKiwi) straightforwardly may not result in plausible explanations. In particular, the OpenKiwi+LIME baseline got higher Recall at Top-K than the random baseline for only 5 LPs. Using randomisation tests with Bonferroni correction, we found that the OpenKiwi+LIME baseline can significantly outperform the random baseline for only 2 LPs (En-Cs and En-Ja). Despite its higher Pearson's correlation at the sentence level, OpenKiwi+LIME yielded random-like (or even worse) explanations for MQM language pairs. This also calls for a stronger baseline for the future edition of the QE shared task. Additional signals/heuristics might be added to the future shared task's baselines such as sparsity of the rationales (as used by IST-Unbabel) and alignments between source and target sentences (as used by HW-TSC and UT-QE).

Critical error detection. By comparing the performance of the submitted systems, in particular the baselines, we see that the difficulty of the *constrained* setting is much higher. We attribute this discrepancy to the fact that the artificially generated data follows a specific set of patterns, which can be captured by current methods when given enough examples. The HyperMT - aiXplain submission seems to be an exception. However, although this system is *unconstrained*, it is composed of fine-tuned decision trees where the base features are *constrained*. We consider that these features are unable to provide sufficient information for the decision trees to be able to identify critical errors, even when fine-tuned on the provided training data.

Due to the scarcity of annotated data containing critical errors, we argue that the *constrained* setting presents a much more realist challenge, where systems are trained for correlating with human judgements but are tested for robustness to critical errors.

For a future edition of this task, we envision a design that simultaneously considers both correlations with human judgements and robustness to critical errors when evaluating a QE system. This can be combined with Task 1, where besides the current evaluation method, participants would also receive a robustness score for their systems, mea-

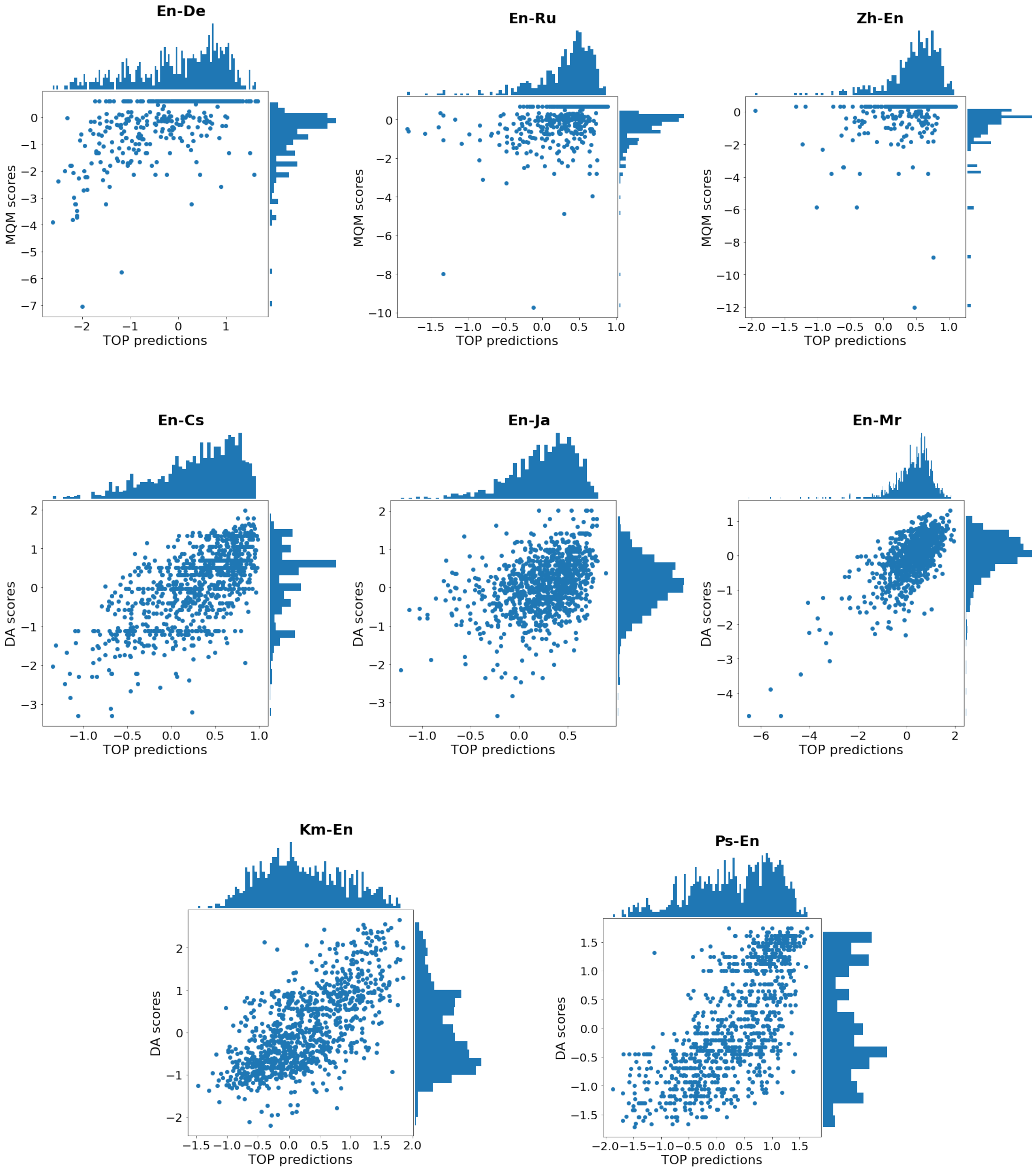


Figure 2: Scatter plots for the predictions against true DA/MQM scores for the top-performing system for each language pair. The histograms show the corresponding marginal distributions of predicted and true scores.

sured on a test set with critical errors. We hope that this configuration would both attract more participants to this task (as it would not required training a specific system for critical error detection) and further motivate the treatment of critical errors in the development of QE systems.

7 Conclusions

This year's edition of the QE Shared Task introduced a number of new elements: new low-resource language pairs (Marathi and Yoruba), new annotation conventions for sentence and word level quality (MQM), new test sets, and new versions of explainability and critical error detection subtasks. The tasks attracted a steady number of participating teams and we believe the overall results are a great reflection of the state-of-the-art in QE.

We have made the gold labels and all submissions to all tasks available for those interested in further analysing the results, while newly interested participants can still access the competition instances on codalab and directly compare their performance to other models. We aspire for the future editions to continue the efforts set in this and previous years and expand the resources and coverage of QE, while further exploring recent and more challenging subtasks such as fine-grained QE, explainable QE and critical error detection.

Acknowledgments

Ricardo Rei and José G. C. de Souza are supported by the P2020 program (MAIA: contract 045909) and by European Union's Horizon Europe Research and Innovation Actions (UTTER: contract 101070631)

André Martins and Chrysoula Zerva are supported by the P2020 program (MAIA: contract 045909), by the European Research Council (ERC StG DeepSPIN 758969), and by the Fundação para a Ciência e Tecnologia through contract UIDB/50008/2020.

Marina Fomicheva and Lucia Specia were supported by funding from the Bergamot project (EU H2020 Grant No. 825303).

References

Hervé Abdi. 2007. The bonferroni and šidák corrections for multiple comparisons. *Encyclopedia of measurement and statistics*, 3:103–107.

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Duarte M. Alves, Ricardo Rei, Ana C. Farinha, José G. C. de Souza, and André F. T. Martins. 2022. Robust MT evaluation with Sentence-level Multilingual data Augmentation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Chantal Amrhein and Rico Sennrich. 2022. Identifying Weaknesses in Machine Translation Metrics Through Minimum Bayes Risk Decoding: A Case Study for COMET. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, Online. Association for Computational Linguistics.

Fatemeh Azadi, Heshaam Faili, and Mohammad Javad Dousti. 2022. Mismatching-Aware Unsupervised Translation Quality Estimation for Low-Resource Languages. *arXiv preprint arXiv:2208.00463*.

Keqin Bao, Yu Wan, Dayiheng Liu, Baosong Yang, Wenqiang Lei, Xiangnan He, Derek F. Wong, and Jun Xie. 2022. Alibaba-translate china's submission for wmt 2022 quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Valerio Basile, Federico Cabitza, Andrea Campagner, and Michael Fell. 2021. Toward a perspectivist turn in ground truthing for predictive computing. *arXiv preprint arXiv:2109.04270*.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. Association for Computational Linguistics.

George Chrysostomou and Nikolaos Aletras. 2022. An empirical study on explanations in out-of-domain settings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6920–6938, Dublin, Ireland. Association for Computational Linguistics.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking Embedding Coupling in Pre-trained Language Models. In *International Conference on Learning Representations*.

Qu Cui, Shujian Huang, Jiahuan Li, Xiang Geng, Zaixiang Zheng, Guoping Huang, and Jiajun Chen. 2021. Directqe: Direct pretraining for machine translation quality estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12719–12727.

David M Eberhard, Gary F Simons, and Charles D Fennig. 2020. Ethnologue: Languages of the world (2020). *URL: https://www. ethnologue. com/(visited on Apr. 11, 2020)(cit. on p. 14)*.

Sugyeong Eo, Chanjun Park, Hyeonseok Moon, Jaehyung Seo, and Heuiseok Lim. 2022. KU X Upstage's submission for the WMT22 Quality Estimation: Critical Error Detection Shared Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Marina Fomicheva, Piyawat Lertvittayakumjorn, Wei Zhao, Steffen Eger, and Yang Gao. 2021. The Eval4NLP shared task on explainable quality estimation: Overview and results. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 165–178, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Erick Fonseca, Chrysoula Zerva, Frédéric Blain, Vishrav Chaudhary, Francisco Guzmán, Nina Lopatina, Lucia Specia, and André F. T. Martins. 2022. MLQE-PE: A multilingual quality estimation and post-editing dataset. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4963–4974, Marseille, France. European Language Resources Association.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021a. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo George Foster, Craig Stewart, Tom Kocmi, Eleftherios Avramidis, Alon Lavie, and André F. T. Martins. 2022. Results of the WMT22 Metrics Shared Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021b. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Xiang Geng, Yu Zhang, Shujian Huang, Shimin Tao, Hao Yang, and Jiajun Chen. 2022. NJUNLP's Participation for the WMT2022 Quality Estimation Shared Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2016. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, FirstView:1–28.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Hui Huang, Hui Di, Chunyou Li, Hanming Wu, Kazushige Oushi, Yufeng Chen, Jian Liu, and Jin'an Xu. 2022. BJTU-Toshiba's Submission to WMT22 Quality Estimation Shared Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Diptesh Kanojia, Marina Fomicheva, Tharindu Ranasinghe, Frédéric Blain, Constantin Orăsan, and Lucia Specia. 2021. Pushing the right buttons: Adversarial evaluation of quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 625–638, Online. Association for Computational Linguistics.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System*

Demonstrations, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 562–568.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Seunghyun S. Lim and Jeonghyeok Park. 2022. Papago's submission to the wmt22 quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Arle Lommel, Aljoscha Burchardt, Maja Popović, Kim Harris, Eleftherios Avramidis, and Hans Uszkoreit. 2014. Using a new analytic measure for the annotation and analysis of mt errors on real data. In *Proceedings of the 17th Annual conference of the European Association for Machine Translation*, pages 165–172.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, USA.

Nikita Mehandru, Marine Carpuat, and Niloufar Selehi. 2022. Quality Estimation by Backtranslation at the WMT 2022 Quality Estimation Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2022. Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

Ricardo Rei, Ana C Farinha, Chrysoula Zerva, Daan van Stigt, Craig Stewart, Pedro Ramos, Taisiya Glushkova, André F. T. Martins, and Alon Lavie. 2021. Are references really needed? unbabel-IST 2021 submission for the metrics shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1030–1040, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C. Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. CometKiwi: IST-Unbabel 2022 Submission for the Quality Estimation Shared Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the WMT 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 684–725, Online. Association for Computational Linguistics.

Chang Su, Miaomiao Ma, Shimin Tao, Hao Yang, Min Zhang, Xiang Geng, Shujian Huang, Jiaxin Guo, Minghan Wang, and Yinglu Li. 2022. CrossQE: HW-TSC 2022 Submission for the Quality Estimation Shared. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Y. Tang, C. Tran, Xian Li, P. Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *ArXiv*, abs/2008.00401.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Yu Wan, Dayiheng Liu, Baosong Yang, Haibo Zhang, Boxing Chen, Derek Wong, and Lidia Chao. 2022. UniTE: Unified translation evaluation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8117–8127, Dublin, Ireland. Association for Computational Linguistics.

Jiayi Wang, Ke Wang, Boxing Chen, Yu Zhao, Weihua Luo, and Yuqi Zhang. 2021. QEMind: Alibaba's submission to the WMT21 quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 948–954, Online. Association for Computational Linguistics.

Evan J. Williams. 1959. *Regression Analysis*, volume 14. Wiley, New York, USA.

Alexander Yeh. 2000. More Accurate Tests for the Statistical Significance of Result Differences. In *Coling-2000: the 18th Conference on Computational Linguistics*, pages 947–953, Saarbrücken, Germany.

Eirini Zafeiridou and Sokratis Sofianopoulos. 2022. Welocalize-ARC/NKUA's Submission to the WMT 2022 Quality Estimation Shared Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

A Official Results of the WMT22 Quality Estimation Task 1 (Direct Assessment)

Tables 9, 10, 11, 12, 13, 14 and 15 show the results for all language pairs and the multilingual variants, ranking participating systems best to worst using Spearman correlation as primary key for each of these cases.

Model	**Spearman**	RMSE	MAE	Disk footprint (B)	# Model params
• IST-Unbabel	0.572	0.689	0.539	2,260,735,089	583,891,109
Papago	0.502	2.404	2.077	2,243,044,839	560,713,447
Welocalize-ARC/NKUA	0.448	0.794	0.632	2,307,101,417	576,733,248
BASELINE	0.415	0.979	0.820	2,280,011,066	564,527,011
lp_sunny ‡	0.414	1.054	0.898	2,356,736,392	580,792,183

Table 9: Official results of the WMT22 Quality Estimation Task 1 **Direct Assessment** for the **Multilingual** variant. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	**Spearman**	RMSE	MAE	Disk footprint (B)	# Model params
• IST-Unbabel	0.605	0.671	0.521	2,260,735,089	583,891,109
Alibaba Translate	0.587	0.675	0.533	2,191,440	560,981,507
Papago	0.571	1.793	1.451	2,243,044,839	560,713,447
Welocalize-ARC/NKUA	0.506	0.733	0.571	2,307,068,585	576,725,041
BASELINE	0.497	0.748	0.585	2,280,011,066	564,527,011
lp_sunny ‡	0.485	0.757	0.596	2,356,736,392	580,792,183

Table 10: Official results of the WMT22 Quality Estimation Task 1 **Direct Assessment** for the **Multilingual (w/o English-Yoruba)** variant. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	**Spearman**	RMSE	MAE	Disk footprint (B)	# Model params
• IST-Unbabel	0.655	0.720	0.545	2,260,735,089	583,891,109
• Papago	0.636	1.371	1.081	2,243,044,839	560,713,447
Alibaba Translate	0.635	0.746	0.607	2,191,440	560,981,507
HW-TSC	0.626	0.712	0.545	540,868,112	222,353,517
Welocalize-ARC/NKUA	0.563	0.785	0.610	2,307,068,585	576,725,041
BASELINE	0.560	0.804	0.608	2,280,011,066	564,527,011
lp_sunny ‡	0.511	0.786	0.614	2,356,736,392	580,792,183
aiXplain	0.477	0.825	0.679	745,679,835	12,345
UCBerkeley-UMD*	0.285	1.252	0.961	–	177,853,440

Table 11: Official results of the WMT22 Quality Estimation Task 1 **Direct Assessment** for the **English-Czech** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. ‡ indicates Codalab usernames of participants from whom we have not received further information and * indicates late submissions that were not considered for the official ranking of participating systems

Model	**Spearman**	RMSE	MAE	Disk footprint (B)	# Model params
• IST-Unbabel	0.385	0.689	0.528	2,260,735,089	583,891,109
Alibaba Translate	0.348	0.673	0.522	2,191,440	560,981,507
HW-TSC	0.341	0.726	0.555	540,868,112	222,353,517
Papago	0.327	2.253	1.957	2,243,044,839	560,713,447
lp_sunny ‡	0.290	0.718	0.556	2,356,736,392	580,792,183
Welocalize-ARC/NKUA	0.276	0.755	0.579	2,307,068,585	576,725,041
aiXplain	0.274	0.704	0.547	745,679,835	12,345
BASELINE	0.272	0.747	0.576	2,280,011,066	564,527,011

Table 12: Official results of the WMT22 Quality Estimation Task 1 **Direct Assessment** for the **English-Japanese** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	**Spearman**	RMSE	MAE	Disk footprint (B)	# Model params
• Papago	0.604	0.658	0.514	2,243,044,839	560,713,447
• Alibaba Translate	0.597	0.456	0.349	2,191,440	560,981,507
• IST-Unbabel	0.592	0.498	0.365	6,932,353,559	583,891,109
• NJUNLP	0.585	0.617	0.414	3,264,730,349	560,145,557
HW-TSC	0.567	0.506	0.372	222,353,517	540,868,112
aiXplain	0.493	0.540	0.396	745,679,835	12,345
Welocalize-ARC/NKUA	0.444	0.534	0.401	2,307,068,585	576,725,041
BASELINE	0.436	0.628	0.461	2,280,011,066	564,527,011
lp_sunny ‡	0.395	0.570	0.443	2,356,736,392	580,792,183

Table 13: Official results of the WMT22 Quality Estimation Task 1 **Direct Assessment** for the **English-Marathi** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	**Spearman**	RMSE	MAE	Disk footprint (B)	# Model params
• IST-Unbabel	0.669	0.714	0.569	2,260,735,089	583,891,109
Alibaba Translate	0.657	0.778	0.596	2,191,440	560,981,507
Papago	0.653	2.786	2.291	2,243,044,839	560,713,447
Welocalize-ARC/NKUA	0.623	0.794	0.619	2,307,068,585	576,725,041
lp_sunny ‡	0.611	0.784	0.621	2,356,736,392	580,792,183
BASELINE	0.579	0.774	0.616	2,280,011,066	564,527,011
HW-TSC	0.509	1.043	0.804	222,353,517	540,868,112

Table 14: Official results of the WMT22 Quality Estimation Task 1 **Direct Assessment** for the **Khmer-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	**Spearman**	RMSE	MAE	Disk footprint (B)	# Model params
● IST-Unbabel	0.722	0.719	0.575	2,260,735,089	583,891,109
Alibaba Translate	0.697	0.720	0.594	2,191,440	560,981,507
Papago	0.671	0.763	0.646	2,243,044,839	560,713,447
HW-TSC	0.661	0.729	0.592	540,868,112	222,353,517
BASELINE	0.641	0.788	0.663	2,280,011,066	564,527,011
lp_sunny ‡	0.637	0.954	0.775	2,356,736,392	580,792,183

Table 15: Official results of the WMT22 Quality Estimation Task 1 **Direct Assessment** for the **Pashto-English** dataset. Teams marked with "●" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. ‡ indicates Codalab usernames of participants from whom we have not received further information.

B Official Results of the WMT22 Quality Estimation Task 1 (MQM)

Tables 16, 17, 18 and 19 show the results for all language pairs and the multilingual variant, ranking participating systems best to worst using Spearman correlation as primary key for each of these cases.

Model	**Spearman**	RMSE	MAE	Disk footprint (B)	# Model params
• IST-Unbabel	0.474	0.973	0.559	2,260,735,089	583,891,109
NJUNLP	0.468	0.945	0.579	3,264,730,349	560,145,557
Alibaba Translate	0.456	0.855	0.493	2,260,733,079	565,137,999
Papago	0.449	1.332	0.990	2,243,044,839	560,713,447
lp_sunny ‡	0.415	0.952	0.536	2,356,736,392	580,792,183
BASELINE	0.317	1.041	0.575	2,280,011,066	564,527,011

Table 16: Official results of the WMT22 Quality Estimation Task 1 **MQM** for the **Multilingual** variant. Baseline systems are highlighted in grey. ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	**Spearman**	RMSE	MAE	Disk footprint (B)	# Model params
• NJUNLP	0.635	0.838	0.594	3,264,730,349	560,145,557
• BJTU-Toshiba	0.621	0.818	0.545	2,239,711,849	559,893,507
pu_nlp ‡	0.611	0.997	0.716	1,326,455,799	237,846,178
Papago	0.582	0.906	0.556	2,243,044,839	560,713,447
IST-Unbabel	0.561	0.854	0.521	2,260,743,851	565,139,485
Alibaba Translate	0.550	0.769	0.466	2,260,733,079	565,137,999
lp_sunny ‡	0.495	0.875	0.534	2,356,736,392	580,792,183
HW-TSC	0.494	0.953	0.612	470,693,617	117,653,760
BASELINE	0.455	0.970	0.576	2,280,011,066	564,527,011
aiXplain	0.376	0.995	0.747	368,857,948	12,345

Table 17: Official results of the WMT22 Quality Estimation Task 1 **MQM** for the **English-German** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	**Spearman**	RMSE	MAE	Disk footprint (B)	# Model params
• IST-Unbabel	0.519	0.963	0.531	2,260,743,915	565,139,485
• Alibaba Translate	0.505	0.961	0.590	2,260,733,079	565,137,999
• Papago	0.496	1.428	1.126	2,243,044,839	560,713,447
• NJUNLP	0.474	0.997	0.666	3,264,730,349	560,145,557
lp_sunny ‡	0.453	0.915	0.548	2,356,736,392	580,792,183
BJTU-Toshiba	0.434	1.011	0.659	2,239,711,849	559,893,507
HW-TSC	0.433	1.257	0.809	2,260,780,823	565,137,436
aiXplain	0.338	1.116	0.785	368,857,948	12,345
BASELINE	0.333	1.051	0.606	2,280,011,066	564,527,011

Table 18: Official results of the WMT22 Quality Estimation Task 1 **MQM** for the **English-Russian** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	**Spearman**	RMSE	MAE	Disk footprint (B)	# Model params
• HW-TSC	0.369	1.163	0.770	2,260,780,823	565,137,436
• IST-Unbabel	0.348	1.073	0.559	2,260,735,089	583,891,109
• Alibaba Translate	0.347	0.989	0.490	2,260,733,079	565,137,999
• Papago	0.325	0.980	0.397	2,243,044,839	560,095,633
• BJTU-Toshiba	0.299	1.128	0.612	1,736,199,083	434,015,235
lp_sunny ‡	0.298	1.064	0.525	2,356,736,392	580,792,183
NJUNLP	0.296	0.999	0.476	3,264,730,349	560,145,557
aiXplain	0.194	1.481	1.079	368,857,948	12,345
BASELINE	0.164	1.102	0.543	2,280,011,066	564,527,011

Table 19: Official results of the WMT22 Quality Estimation Task 1 **MQM** for the **Chinese-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey. ‡ indicates Codalab usernames of participants from whom we have not received further information.

C Official Results of the WMT22 Quality Estimation Task 1 (Word-level)

Tables 20, 21, 22, 23, 24, 25, 26, 27, 28 and 29 show the results for all language pairs and the multilingual variants, ranking participating systems best to worst using Matthews correlation coefficient (MCC) as primary key for each of these cases.

Model	**MCC**	Recall	Precision	Disk footprint (B)	# Model params
• IST-Unbabel	0.341	0.466	0.810	2,260,744,555	565,139,485
Papago	0.317	0.422	0.787	2,241,394,304	560,301,035
BASELINE	0.235	0.356	0.765	2,280,011,066	564,527,011

Table 20: Official results of the WMT22 Quality Estimation Task 1 (word-level) for the **Multilingual** task. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey.

Model	**MCC**	Recall	Precision	Disk footprint (B)	# Model params
• IST-Unbabel	0.361	0.494	0.830	2,260,744,555	565,139,485
• Papago	0.343	0.451	0.858	2,241,394,304	560,301,035
BASELINE	0.257	0.378	0.838	2,280,011,066	564,527,011
HW-TSC	0.218	0.404	0.628	2,336,352,552	612,368,384

Table 21: Official results of the WMT22 Quality Estimation Task 1 (word-level) for the **Multilingual w/o English-Yoruba** task. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey.

Model	**MCC**	Recall	Precision	Disk footprint (B)	# Model params
• IST-Unbabel	0.436	0.578	0.852	2,260,744,555	565,139,485
• HW-TSC	0.424	0.570	0.848	2,260,780,823	565,137,436
• Papago	0.396	0.549	0.739	2,240,570,795	560,095,834
BASELINE	0.325	0.426	0.870	2,280,011,066	564,527,011

Table 22: Official results of the WMT22 Quality Estimation Task 1 (word-level) for the **English-Czech** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Model	**MCC**	Recall	Precision	Disk footprint (B)	# Model params
• HW-TSC	0.258	0.497	0.728	2,260,780,823	565,137,436
• Papago	0.257	0.502	0.699	2,241,394,304	560,301,035
• IST-Unbabel	0.238	0.491	0.687	2,260,743,979	565,139,485
BASELINE	0.175	0.375	0.795	2,280,011,066	564,527,011

Table 23: Official results of the WMT22 Quality Estimation Task 1 (word-level) for the **English-Japanese** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey. “Rank” indicates the averaged ranking of participants with regards to all metrics (including memory print and number of parameters) and not the final shared task ranking which is decided according to MCC.

Model	**MCC**	Recall	Precision	Disk footprint (B)	# Model params
• Papago	0.418	0.420	0.951	2,241,394,304	560,301,035
• NJUNLP	0.412	0.472	0.939	3,264,730,349	560,145,557
• IST-Unbabel	0.392	0.414	0.947	2,260,744,107	565,139,485
HW-TSC	0.351	0.428	0.917	2,260,780,823	565,137,436
BASELINE	0.306	0.282	0.946	2,280,011,066	564,527,011

Table 24: Official results of the WMT22 Quality Estimation Task 1 (word-level) for the **English-Marathi** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Model	**MCC**	Recall	Precision	Disk footprint (B)	# Model params
• Papago	0.429	0.762	0.660	2,241,394,304	560,301,035
• IST-Unbabel	0.425	0.779	0.555	2,260,744,107	565,139,485
• NJUNLP	0.421	0.744	0.677	3,264,730,349	560,145,557
BASELINE	0.402	0.769	0.567	2,280,011,066	564,527,011
HW-TSC	0.353	0.759	0.395	2,260,780,823	565,137,436

Table 25: Official results of the WMT22 Quality Estimation Task 1 (word-level) for the **Khmer-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Model	**MCC**	Recall	Precision	Disk footprint (B)	# Model params
• IST-Unbabel	0.424	0.691	0.733	2,260,744,107	565,139,485
Papago	0.374	0.646	0.723	2,241,394,304	560,301,035
BASELINE	0.359	0.695	0.628	2,280,011,066	564,527,011
HW-TSC	0.358	0.699	0.597	2,260,780,823	565,137,436

Table 26: Official results of the WMT22 Quality Estimation Task 1 (word-level) for the **Pashto-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Model	**MCC**	Recall	Precision	Disk footprint (B)	# Model params
• NJUNLP	0.352	0.351	0.980	3,264,730,349	560,145,557
• Papago	0.319	0.336	0.960	2,241,394,304	560,301,035
• IST-Unbabel	0.303	0.317	0.956	2,260,744,107	565,139,485
HW-TSC	0.274	0.292	0.954	2,260,780,823	565,137,436
BASELINE	0.182	0.213	0.970	2,280,011,066	564,527,011

Table 27: Official results of the WMT22 Quality Estimation Task 1 (word-level) for the **English-German** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Model	**MCC**	Recall	Precision	Disk footprint (B)	# Model params
• IST-Unbabel	0.427	0.468	0.958	2,260,743,915	565,139,485
• Papago	0.421	0.381	0.966	2,241,394,304	560,713,447
• NJUNLP	0.390	0.440	0.949	3,264,730,349	560,145,557
HW-TSC	0.343	0.396	0.945	2,260,780,823	565,137,436
BASELINE	0.203	0.144	0.960	2,280,011,066	564,527,011

Table 28: Official results of the WMT22 Quality Estimation Task 1 (word-level) for the **English-Russian** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Model	**MCC**	Recall	Precision	Disk footprint (B)	# Model params
• IST-Unbabel	0.360	0.327	0.966	2,260,743,915	565,139,485
• Papago	0.351	0.338	0.973	2,241,394,304	560,713,447
• NJUNLP	0.308	0.303	0.988	3,264,730,349	560,145,557
HW-TSC	0.246	0.181	0.910	2,260,780,823	565,137,436
BASELINE	0.104	0.123	0.965	2,280,011,066	564,527,011

Table 29: Official results of the WMT22 Quality Estimation Task 1 (word-level) for the **Chinese-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

D Official Results of the WMT22 Quality Estimation Task 2 (Explainable QE)

Tables 30, 31, 32, 33, 34, 35, 36, 37 and 38 show the results for all language pairs, ranking participating systems best to worst using "Recall at Top-K" on target sentences as primary key for each of these cases.

Model	Word-level (Target sentence)			Sentence-level	
	Recall at Top-K	AUC	AP	Pearson's	Spearman's
• IST-Unbabel	0.561	0.725	0.659	0.548	0.511
• HW-TSC	0.536	0.709	0.632	0.314	0.323
BASELINE (OpenKiwi+LIME)	0.417	0.537	0.500	0.342	0.352
BASELINE (Random)	0.363	0.493	0.453	0.011	0.016

Table 30: Official results of the WMT22 Quality Estimation Task 2 for the **English-Czech** dataset. Teams marked with "•" correspond to the winning submissions, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Model	Word-level (Target sentence)			Sentence-level	
	Recall at Top-K	AUC	AP	Pearson's	Spearman's
• IST-Unbabel	0.466	0.641	0.557	0.252	0.243
• HW-TSC	0.462	0.651	0.547	0.132	0.148
BASELINE (OpenKiwi+LIME)	0.367	0.509	0.451	0.202	0.217
BASELINE (Random)	0.336	0.503	0.418	0.028	0.019

Table 31: Official results of the WMT22 Quality Estimation Task 2 for the **English-Japanese** dataset. Teams marked with "•" correspond to the winning submissions, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Model	Word-level (Target sentence)			Sentence-level	
	Recall at Top-K	AUC	AP	Pearson's	Spearman's
• IST-Unbabel	0.317	0.667	0.448	0.585	0.467
• HW-TSC	0.280	0.625	0.412	0.317	0.426
BASELINE (OpenKiwi+LIME)	0.194	0.479	0.310	0.336	0.372
BASELINE (Random)	0.167	0.489	0.296	0.043	0.017

Table 32: Official results of the WMT22 Quality Estimation Task 2 for the **English-Marathi** dataset. Teams marked with "•" correspond to the winning submissions, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Model	Word-level (Target sentence)			Sentence-level	
	Recall at Top-K	AUC	AP	Pearson's	Spearman's
• IST-Unbabel	0.390	0.747	0.511	0.416	0.459
HW-TSC	0.313	0.686	0.422	0.369	0.426
BASELINE (Random)	0.148	0.527	0.256	0.022	0.015
BASELINE (OpenKiwi+LIME)	0.135	0.428	0.230	0.252	0.330

Table 33: Official results of the WMT22 Quality Estimation Task 2 for the **English-Russian** dataset. Teams marked with "•" correspond to the winning submissions, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Model	Word-level (Target sentence)			Sentence-level	
	Recall at Top-K	AUC	AP	Pearson's	Spearman's
• IST-Unbabel	0.365	0.776	0.490	0.559	0.553
HW-TSC	0.252	0.689	0.361	0.375	0.435
BASELINE (Random)	0.124	0.504	0.212	-0.049	-0.043
BASELINE (OpenKiwi+LIME)	0.074	0.442	0.172	0.370	0.414

Table 34: Official results of the WMT22 Quality Estimation Task 2 for the **English-German** dataset. Teams marked with "•" correspond to the winning submissions, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Model	Word-level (Target sentence)			Sentence-level	
	Recall at Top-K	AUC	AP	Pearson's	Spearman's
• IST-Unbabel	0.234	0.671	0.359	0.309	0.321
BASELINE (Random)	0.144	0.514	0.246	-0.086	-0.101
BASELINE (OpenKiwi+LIME)	0.111	0.442	0.218	0.085	0.160

Table 35: Official results of the WMT22 Quality Estimation Task 2 for the **English-Yoruba** dataset. Teams marked with "•" correspond to the winning submissions, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Model	Word-level (Target sentence)			Sentence-level	
	Recall at Top-K	AUC	AP	Pearson's	Spearman's
• HW-TSC	0.686	0.720	0.751	0.601	0.610
IST-Unbabel	0.665	0.660	0.751	0.617	0.598
UT-QE	0.622	0.628	0.694	0.222	0.190
BASELINE (OpenKiwi+LIME)	0.580	0.520	0.653	0.417	0.430
BASELINE (Random)	0.565	0.498	0.633	-0.048	-0.045

Table 36: Official results of the WMT22 Quality Estimation Task 2 for the **Khmer-English** dataset. Teams marked with "•" correspond to the winning submissions, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Model	Word-level (Target sentence)			Sentence-level	
	Recall at Top-K	AUC	AP	Pearson's	Spearman's
• HW-TSC	0.715	0.716	0.777	0.393	0.418
IST-Unbabel	0.672	0.612	0.740	0.593	0.601
UT-QE	0.668	0.643	0.727	0.409	0.402
BASELINE (OpenKiwi+LIME)	0.615	0.503	0.676	0.378	0.403
BASELINE (Random)	0.614	0.497	0.662	-0.002	0.002

Table 37: Official results of the WMT22 Quality Estimation Task 2 for the **Pashto-English** dataset. Teams marked with "•" correspond to the winning submissions, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Model	Word-level (Target sentence)			Sentence-level	
	Recall at Top-K	AUC	AP	Pearson's	Spearman's
• IST-Unbabel	0.379	0.785	0.475	0.103	0.190
HW-TSC	0.220	0.652	0.315	0.097	0.159
BASELINE (Random)	0.093	0.463	0.162	0.041	-0.010
BASELINE (OpenKiwi+LIME)	0.048	0.388	0.126	-0.007	0.159

Table 38: Official results of the WMT22 Quality Estimation Task 2 for the **Chinese-English** dataset. Teams marked with "•" correspond to the winning submissions, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

E Official Results of the WMT22 Quality Estimation Task 3 (Critical Error Detection)

Tables 39, 40, 41 and 42 show the results for all language pairs and the multilingual variants, ranking participating systems best to worst using Matthews correlation coefficient (MCC) as primary key for each of these cases.

Model	**MCC**	Recall	Precision	Disk footprint (B)	# Model params
• IST-Unbabel	0.564	0.619	0.619	2,260,735,025	565,137,435
BASELINE	0.074	0.191	0.191	2,277,430,785	569,330,715

Table 39: Official results of the WMT22 Quality Estimation Task 3 (Critical Error Detection) for the **English-German (Constrained)** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Model	**MCC**	Recall	Precision	Disk footprint (B)	# Model params
• KU X Upstage	0.964	0.968	0.968	2,244,861,551	559,890,432
BASELINE	0.855	0.873	0.873	2,260,734,129	565,137,435
aiXplain	0.219	0.318	0.318	2,052,963,739	12,345

Table 40: Official results of the WMT22 Quality Estimation Task 3 (Critical Error Detection) for the **English-German (UNconstrained)** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Model	**MCC**	Recall	Precision	Disk footprint (B)	# Model params
• IST-Unbabel	0.721	0.761	0.761	2,260,735,025	565,137,435
BASELINE	-0.001	0.141	0.141	2,277,430,785	569,330,715

Table 41: Official results of the WMT22 Quality Estimation Task 3 (Critical Error Detection) for the **Portuguese-English (Constrained)** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Model	**MCC**	Recall	Precision	Disk footprint (B)	# Model params
• KU X Upstage	0.984	0.986	0.986	2,244,861,551	559,890,432
BASELINE	0.934	0.944	0.944	2,260,734,129	565,137,435
aiXplain	0.179	0.296	0.296	9,395,107	12,345

Table 42: Official results of the WMT22 Quality Estimation Task 3 (Critical Error Detection) for the **Portuguese-English (UNconstrained)** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system based on randomisation tests with Bonferroni correction (Yeh, 2000). Baseline systems are highlighted in grey.

Findings of the WMT 2022 Shared Task on Efficient Translation

Kenneth Heafield Biao Zhang Graeme Nail Jelmer van der Linde Nikolay Bogoychev
University of Edinburgh
10 Crichton Street
Edinburgh, Scotland EH8 9AB
{Kenneth.Heafield,b.zhang,graeme.nail,jelmer.vanderlinde,n.bogoych}@ed.ac.uk

Abstract

The machine translation efficiency task challenges participants to make their systems faster and smaller with minimal impact on translation quality. How much quality to sacrifice for efficiency depends upon the application, so participants were encouraged to make multiple submissions covering the space of trade-offs. In total, there were 76 submissions from 5 teams. The task covers GPU, single-core CPU, and multi-core CPU hardware tracks as well as batched throughput or single-sentence latency conditions. Submissions showed hundreds of millions of words can be translated for a dollar, average latency is 3.5–25 ms, and models fit in 7.5–900 MB.

1 Introduction

The efficiency task complements the collocated news task by challenging participants to make their machine translation systems computationally efficient. This is the fifth edition of the task, expanding upon previous editions (Heafield et al., 2021, 2020; Hayashi et al., 2019; Birch et al., 2018).

Participants built English→German machine translation systems following a constrained data condition. The data condition follows the constrained **2021** Workshop on Machine Translation news translation task. This year, to reduce the barrier to entry, organisers provided an ensemble of teacher systems, as well as cleaned data and distilled output from the teacher ensemble. Participants were required to use the provided teacher systems, but were free to distil additional data from the constrained condition. The SentencePiece vocabulary used by the teachers was also made available.

For translation quality measurement, we use the news-focused WMT22 dataset, and the systems are ranked according to the COMET (Rei et al., 2020) automatic metric. We also evaluate systems on BLEU and chrF for additional reference.

	Throughput		**Latency**	
	CPU-ALL	GPU	CPU-1	GPU
CUNI	1	1	1	1
ECNU	1	1	1	1
Edinburgh	15	11	15	11
HuaweiTSC	5		5	
RoyalFlush				6

Table 1: Number of systems submitted by each participant for the different hardware and batching conditions. CPU-ALL refers to the 36-core hardware setting.

Submissions are made as Docker containers so we can consistently measure their performance in terms of quality, speed, memory usage, and disk space. We run the containers in three different hardware environments: one GPU, one CPU core, and multiple CPU cores. Systems were tested for throughput by providing 1 million sentences upfront to allow batching and parallelization. We also tested for latency with a program that drip-feeds one input sentence, waits for the translation, and then provides the next input sentence. There were four conditions in total: GPU throughput, GPU Latency, 1 CPU Core Latency, and 36 CPU cores throughput. We did not measure latency in a multi-core CPU setting because the test hardware has 36 cores and overhead for 36 threads is larger than the cost of arithmetic for the small tensors in optimized models. We also did not measure throughput on a single CPU core as we found that setting to be a somewhat unrealistic real world scenario.

Participants were free to choose which conditions to participate in. The condition was passed to the Docker container as command line arguments. Table 1 shows the five participants and the number of systems they submitted to each of the conditions.

Machine translation is used in a range of settings where users might choose different trade-offs between quality and efficiency. For example, a high-frequency trading system might prefer the lowest latency at the expense of quality given that the out-

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 100 - 108
December 7-8, 2022 ©2022 Association for Computational Linguistics

put will only be read by a machine. Conversely, in a post-editing scenario the personnel costs outweigh many computational costs. Therefore there is not a single best system, but a range of options that trade between quality and efficiency.

We emphasize the Pareto frontier: the fastest systems at each level of quality, or the smallest systems at each level of quality. To explore the Pareto frontier, participants were encouraged to make multiple submissions covering the range of trade-offs. In total, 76 combinations of models, hardware, and batching were benchmarked.

2 Hardware

We chose modern hardware to encourage exploiting new hardware features. The GPU is an NVidia A100 from the Oracle Cloud `BM.GPU4.8` instance. The instance has eight GPUs and we limited Docker to using only one GPU. The GPU machine has an AMD EPYC 7542 CPU with all cores allowed.

The CPU-only condition used a dual-socket Intel Xeon Gold 6354 from Oracle Cloud `BM.Optimized3.36` with a total of 36 cores. For the single-core CPU track, we reserved the entire machine then ran Docker with `-cpuset-cpus=0`. In the 36-core CPU track, participants were free to configure their own CPU sets and affinities.

The Oracle Cloud machines are bare metal servers, meaning there was no shared tenancy, no virtualization, and the test machines were otherwise quiescent.

3 Input Text

To amortize loading time, avoid starving highly parallel submissions, and reduce the ability to cheat, we benchmark systems on 1 million sentences of input. The test set is hidden inside these 1 million sentences, shuffled with filler sentences. Many filler sentences are drawn from parallel corpora to check that systems are in fact translating all sentences, though we do not consider scores on noisy corpora reliable enough to report. The composition of this set changes each year and is decided after the submission deadline.

The filler data was gathered from parallel corpora and gender bias challenge sets: WMT news test sets from 2008 through 2022 (Akhbardeh et al., 2021), the additional test inputs in WMT 2021, Khresmoi summary test v2 (Dušek et al., 2017), IWSLT 2019 (Jan et al., 2019), SimpleGen (Renduchintala et al., 2021), WinoMT (Stanovsky et al., 2019), TED 2020 (Reimers and Gurevych, 2020), and Tilde RAPID 2019 (Rozis and Skadiņš, 2017). We limit sentence lengths to 150 space-separated tokens. Because WMT 2020 includes excessively long segments that are actually concatenated sentences, we also added sentence split versions of WMT 2020 and WMT 2021, though the difference on WMT 2021 was minor. Source sentences were concatenated, deduplicated, and shuffled. The Tilde RAPID corpus was clipped to make a total of 1 million deduplicated lines. Counts are shown in Table 2.

Corpus	**Sentences**
WMT 08–19	32,477
WMT 20 under 150 tokens	1,416
WMT 20 sentence split	2,048
WMT 21 sentence split	1,096
WMT 21 inc. additional tests	14,938
WMT 22	2,037
Khresmoi Summary Test v2	1,000
IWSLT 2019	2,278
SimpleGen	2,664
WinoMT	3,888
TED 2020 v1	293,562
Tilde RAPID 2019	663,922
Total	1,021,326
Deduplicated	1,000,000

Table 2: Summary of corpora used for the input text.

Input text and tools to extract test sets from system outputs are available at `https://data.statmt.org/wmt22/efficiency-task/wmt22-testdata.tar.xz`.

The input file is 1,000,000 lines, consisting of 19,926,744 space-separated words, or 124,186,772 bytes of English text in UTF-8. This is a mean of 19.9 words per sentence and is comparable to the previous year (Heafield et al., 2021). Teams were responsible for their own tokenization and detokenization; for this they were permitted to use the SentencePiece vocabulary provided with the teacher system, or to implement an alternative. We provided raw UTF-8 English input text with one sentence per line.

4 Metrics

4.1 Resources

Time was measured with wall (real) time reported by `time` and CPU time reported by the kernel for the process group. We do not measure loading time because it is small compared to translating 1 million sentences, some tools load lazily, and it is easily gamed by padding loading time.

Peak RAM consumption was measured using `memory.max_usage` in bytes from the kernel for the CPU and by polling `nvidia-smi` for the GPU. Swap was disabled.

Participants were instructed to separate their Docker images into model and code files so that models could be measured separately from the relatively noisy size of code and libraries. A model was defined as "everything derived from data: all model parameters, vocabulary files, BPE configuration if applicable, quantization parameters or lookup tables where applicable, and hyperparameters like embedding sizes." Code could include "simple rule-based tokenizer scripts and hard-coded model structure that could plausibly be used for another language pair." They were also permitted to use standard compression tools such as `xz` to compress models; decompression time was excluded in results. We report size of the model directory captured before the model ran. We also measured the total size of the Docker image (after compressing with `xz`).

4.2 Quality

Translation quality is measured on the WMT 2022 news test set. The automatic metrics are COMET (Rei et al., 2020) from `unbabel-comet` version `1.1.3` with the pretrained model `wmt20-comet-da`, BLEU from sacrebleu (Post, 2018) `nrefs:1|case:mixed|eff:no|tok:13a |smooth:exp|version:2.3.1`, and chrF also from sacrebleu.

5 Results

The results of the task evaluation for the latency scenario are presented in Table 3, and those for throughput are presented in Table 4. Results are separated by the different hardware conditions and within each hardware setting the results are ordered by their COMET score, which is shown to have closer correspondence to human evaluation as compared to BLEU and ChrF (Freitag et al., 2021).

Figure 1 shows the trade-off between quality and speed of batched translation submissions separated by hardware environment. Each plot shows the Pareto frontier as a black staircase to highlight the best combinations of quality and speed. While GPU systems (Figure 1a) achieve higher throughput compared to CPU systems (Figure 1b), this ignores pricing differences between these compute options. In Figure 2, we combine GPU and 36 Core CPU speed by using Oracle Cloud pricing. Despite the less expensive per-hour pricing of CPU, GPU is cheaper for throughput-oriented tasks that allow batching.

The all-hardware latency Pareto frontier is shown in Figure 3. This year all participants submitted systems to the latency task. This year, for the first time, the semi-autoregressive GPU system by RoyalFlush dominates the lower quality settings of the latency Pareto frontier, with Edinburgh GPU systems having won on some higher quality systems.

Model sizes at rest on disk appear in Figures 4a. Participants were allowed to compress their models using their own tools and standard tools like `xz`. The Pareto frontier consists of almost entirely Edinburgh submissions, with HuaweiTSC producing several systems on the lower quality settings, due to their 4-bit compression models. Docker image sizes, which include model and software, appear in Figure 4b, where the Pareto frontier is dominated by Edinburgh submissions. Conversely, some others opted to optimize other metrics and included large Linux installations. We compressed all docker images with `xz` before measuring.

Memory (RAM) consumption appears in Figure 5. GPU memory consumption reflects batch size and some participants set a large batch size to maximize speed. Optimizing speed for multi-socket CPU machines implies having a copy of the model in RAM close to each socket, so memory consumption is larger beyond simply having temporary space for more batches. Finally, participants may have sorted the entire 118 MB input file in RAM to form batches of equal length sentences. RoyalFlush is the clear winner on the GPU latency RAM consumption, and HuaweiTSC is the winner of CPU latency RAM consumption.

6 Conclusion

Using the highest quality system in this evaluation, translating 124,186,772 characters took 283

NVIDIA A100 GPU Latency

		Automatic			Seconds		Disk MB		RAM MB
Team	Variant	COMET	BLEU	chrF	Wall	CPU	Model	Docker	GPU
Edinburgh	6-1.base.wide-gpu	0.542	34.50	61.90	15051	15141	900	2316	37961
Edinburgh	12_1.large-gpu	0.541	34.10	61.60	14116	14186	624	2039	37555
Edinburgh	6-2.base-gpu	0.528	33.80	61.50	16548	16584	171	1587	37181
Edinburgh	12_1.base-gpu	0.518	33.90	61.40	13081	13118	225	1641	37211
RoyalFlush	royalflush_hrt_e20d1_k2	0.512	33.80	61.50	6008	6051	345	869	2021
Edinburgh	6-1.base-gpu	0.507	33.50	61.10	12665	12698	159	1574	37175
RoyalFlush	royalflush_hrt_e12d1_k2	0.498	33.90	61.40	5437	5472	259	781	1973
Edinburgh	8-4.tied.tiny-gpu	0.462	32.40	60.10	24126	24157	84	1500	37133
RoyalFlush	royalflush_hrt_e20d1_k3	0.458	33.40	61.10	4706	4752	345	870	2021
Edinburgh	6-2.micro.4h-gpu	0.454	31.70	59.80	15003	15031	74	1489	37129
Edinburgh	6-2.tied.tiny-gpu	0.443	31.50	59.50	15236	15261	77	1492	37129
ECNU	ecnu-mt	0.432	33.20	60.70	25306	25338	492	15680	4989
Edinburgh	6-2.micro.1h-gpu	0.432	31.30	59.20	14789	14817	73	1489	37129
RoyalFlush	royalflush_hrt_e12d1_k3	0.430	33.30	60.90	4093	4129	257	783	1973
Edinburgh	ib-6-2-tiny-gpu	0.388	31.10	59.40	12624	12653	81	1496	37133
RoyalFlush	royalflush_hrt_e20d1_k4	0.376	33.00	60.80	4024	4064	343	866	2021
Edinburgh	ib-12_1-tiny-gpu	0.373	31.90	59.80	10763	10793	99	1515	37141
RoyalFlush	royalflush_hrt_e12d1_k4	0.342	32.60	60.30	3409	3443	259	783	1973
CUNI	cuni-large-ende	0.250	30.80	59.10	8327	8410	856	1676	1875

1 Core Ice Lake CPU Latency

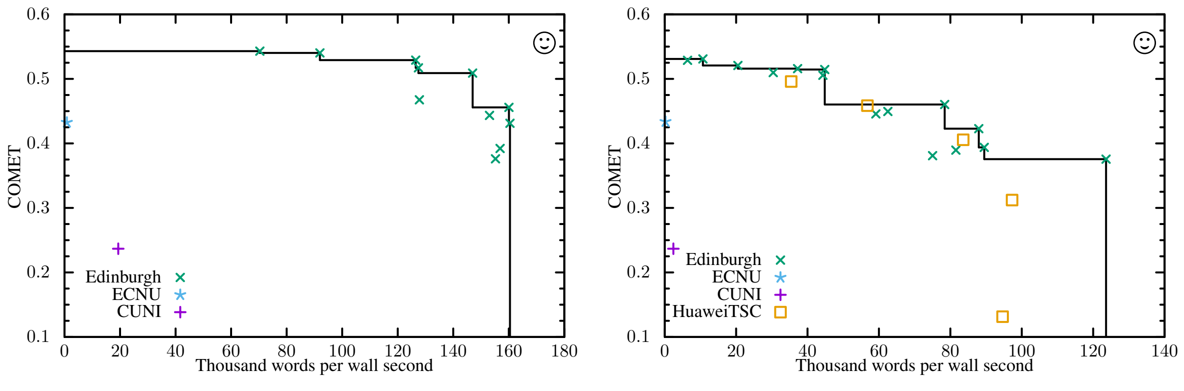
		Automatic			Seconds		Disk MB		RAM MB
Team	Variant	COMET	BLEU	chrF	Wall	CPU	Model	Docker	CPU
Edinburgh	6-1.base.wide-cpu	0.517	33.90	61.50	79230	79234	162	212	2487
Edinburgh	12_1.large-cpu	0.516	33.70	61.30	51991	51995	121	171	1537
Edinburgh	12_1.base_efh_0.05	0.513	33.80	61.40	37183	37190	176	1176	1337
Edinburgh	6-2.base-cpu	0.509	33.30	61.00	18101	18102	32	82	542
Edinburgh	12_1.base_efh_0.05_ft8	0.507	33.50	61.20	14669	14679	156	217	1256
Edinburgh	6-1.base-cpu	0.496	33.10	60.90	13383	13385	29	79	533
Edinburgh	12_1.base-cpu	0.494	33.70	61.20	19100	19102	44	94	640
HuaweiTSC	huawei.cpu.base.docker	0.485	34.00	61.10	15743	15741	40	112	254
HuaweiTSC	huawei.cpu.sm.docker	0.455	32.90	60.30	9955	9954	22	94	162
Edinburgh	8-4.tied.tiny_efh_0.3_ft8	0.444	31.80	59.70	13360	13361	36	97	459
Edinburgh	ib-12-4-micro-cpu	0.442	31.90	59.90	12071	12072	18	68	328
Edinburgh	8-4.tied.tiny-cpu	0.439	31.60	59.60	14090	14090	15	65	270
ECNU	ecnu-mt	0.434	33.20	60.70	327823	327764	492	14469	4900
Edinburgh	6-2.micro.4h-cpu	0.418	30.90	59.20	8916	8917	13	63	247
HuaweiTSC	huawei.cpu.t12.docker	0.417	32.20	59.70	7591	7590	15	87	122
Edinburgh	6-2.micro.1h-cpu	0.383	29.90	58.40	8632	8632	13	63	256
Edinburgh	6-2.tied.tiny-cpu	0.378	30.00	58.50	9371	9372	13	63	257
Edinburgh	ib-6-3-tiny-cpu	0.372	30.40	58.80	9258	9258	15	65	302
Edinburgh	12_1.tiny_efh_0.5_ft8	0.371	30.00	58.50	6590	6592	30	91	374
HuaweiTSC	huawei.cpu.t6.docker	0.315	30.20	58.30	5871	5870	11	84	100
CUNI	cuni-large-ende	0.250	30.80	59.10	335787	335806	856	1676	4857
HuaweiTSC	huawei.cpu.ex.docker	0.128	26.30	55.10	6286	6285	7	80	70

Table 3: Results of system evaluation on the latency task. Total time measured in seconds is equivalent to microseconds/sentence because the input is 1 million sentences.

NVIDIA A100 GPU Batch									
		Automatic			Seconds		Disk MB		RAM MB
Team	Variant	COMET	BLEU	chrF	Wall	CPU	Model	Docker	GPU
Edinburgh	6-1.base.wide-gpu	0.543	34.60	61.90	283	349	900	2316	37961
Edinburgh	12_1.large-gpu	0.540	34.10	61.60	217	262	624	2039	37555
Edinburgh	6-2.base-gpu	0.529	33.80	61.50	158	169	171	1587	37181
Edinburgh	12_1.base-gpu	0.517	33.90	61.50	156	172	225	1641	37211
Edinburgh	6-1.base-gpu	0.509	33.40	61.10	136	146	159	1574	37175
Edinburgh	8-4.tied.tiny-gpu	0.468	32.50	60.20	156	161	84	1500	37133
Edinburgh	6-2.micro.4h-gpu	0.456	31.90	59.90	125	128	74	1489	37129
Edinburgh	6-2.tied.tiny-gpu	0.443	31.50	59.50	130	134	77	1492	37129
ECNU	ecnu-mt	0.432	33.20	60.70	23600	23643	492	15680	5719
Edinburgh	6-2.micro.1h-gpu	0.431	31.30	59.20	124	128	73	1489	37129
Edinburgh	ib-6-2-tiny-gpu	0.392	31.10	59.50	127	132	81	1496	37133
Edinburgh	ib-12_1-tiny-gpu	0.376	32.10	59.90	128	134	99	1515	37141
CUNI	cuni-large-ende	0.237	30.80	59.10	1029	1115	856	1676	4179

36 Core Ice Lake CPU Batch									
		Automatic			Seconds		Disk MB		RAM MB
Team	Variant	COMET	BLEU	chrF	Wall	CPU	Model	Docker	CPU
Edinburgh	12_1.large-cpu	0.531	33.90	61.40	1864	65214	121	171	57879
Edinburgh	6-1.base.wide-cpu	0.529	34.10	61.60	3121	108057	162	212	77379
Edinburgh	12_1.base_efh_0.05	0.521	34.00	61.50	972	34532	176	1176	32754
Edinburgh	6-2.base-cpu	0.516	33.50	61.20	535	18982	32	82	24467
Edinburgh	12_1.base_efh_0.05_ft8	0.514	33.70	61.40	445	15571	156	217	22373
Edinburgh	12_1.base-cpu	0.510	34.00	61.40	656	23159	44	94	33434
Edinburgh	6-1.base-cpu	0.506	33.30	61.00	450	15795	29	79	23520
HuaweiTSC	huawei.cpu.base.docker	0.496	34.10	61.30	562	36577	40	112	17513
Edinburgh	8-4.tied.tiny_efh_0.3_ft8	0.460	31.90	59.80	254	8909	36	97	16473
HuaweiTSC	huawei.cpu.sm.docker	0.459	32.90	60.30	351	21437	22	94	12461
Edinburgh	8-4.tied.tiny-cpu	0.450	31.90	59.80	319	11041	15	65	13880
Edinburgh	ib-12-4-micro-cpu	0.446	32.00	60.00	337	11781	18	68	16707
ECNU	ecnu-mt	0.434	33.20	60.70	88463	2059785	492	14469	2103
Edinburgh	6-2.micro.4h-cpu	0.423	30.90	59.30	227	7925	13	63	11154
HuaweiTSC	huawei.cpu.t12.docker	0.406	31.80	59.60	238	13532	15	87	5797
Edinburgh	6-2.micro.1h-cpu	0.394	30.00	58.50	223	7671	13	63	10526
Edinburgh	6-2.tied.tiny-cpu	0.390	30.30	58.50	244	8559	13	63	12804
Edinburgh	ib-6-3-tiny-cpu	0.381	30.50	58.90	266	9280	15	65	13464
Edinburgh	12_1.tiny_efh_0.5_ft8	0.376	30.20	58.60	161	5531	30	91	11843
HuaweiTSC	huawei.cpu.t6.docker	0.312	30.20	58.40	205	11147	11	84	7166
CUNI	cuni-large-ende	0.237	30.80	59.10	8243	295751	856	1676	138539
HuaweiTSC	huawei.cpu.ex.docker	0.131	26.20	55.20	211	11495	7	80	7458

Table 4: Results of system evaluation on the throughput task. Total time measured in seconds is equivalent to microseconds/sentence because the input is 1 million sentences.



(a) Speed on GPU with COMET for all systems

(b) Speed on 36 Cores with COMET for all systems

Figure 1: Speed and quality of batched submissions. The staircase shows the Pareto frontier.

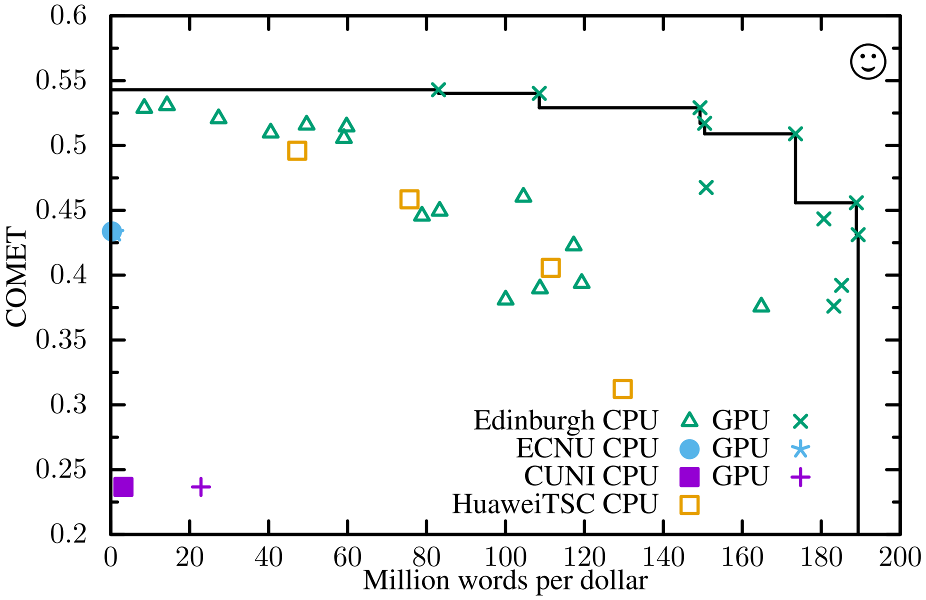


Figure 2: Cost of batched translation for an A100 GPU at $3.05/hr or 36 Cores of CPU at $2.7/hr on Oracle Cloud. For readability, we omit systems with a COMET score less than 0.2.

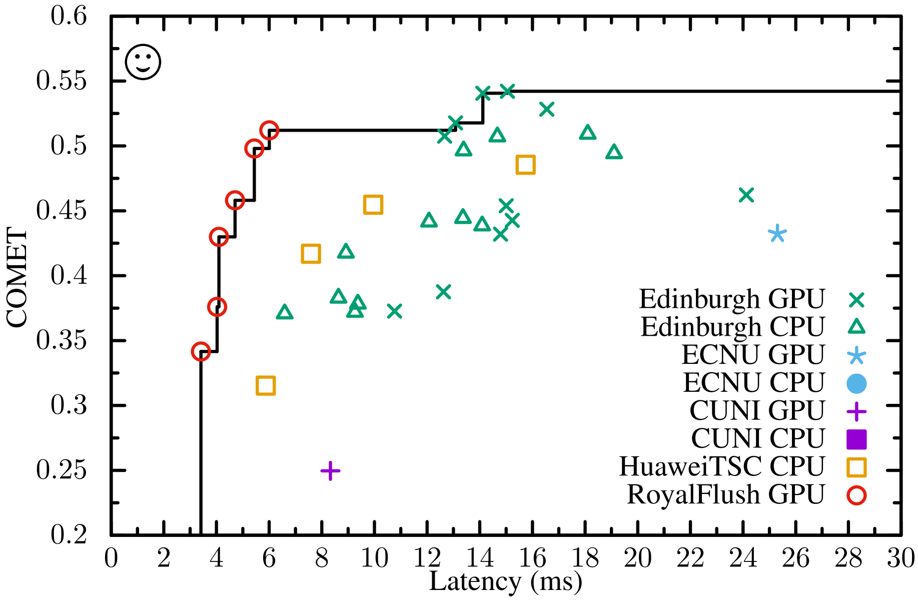
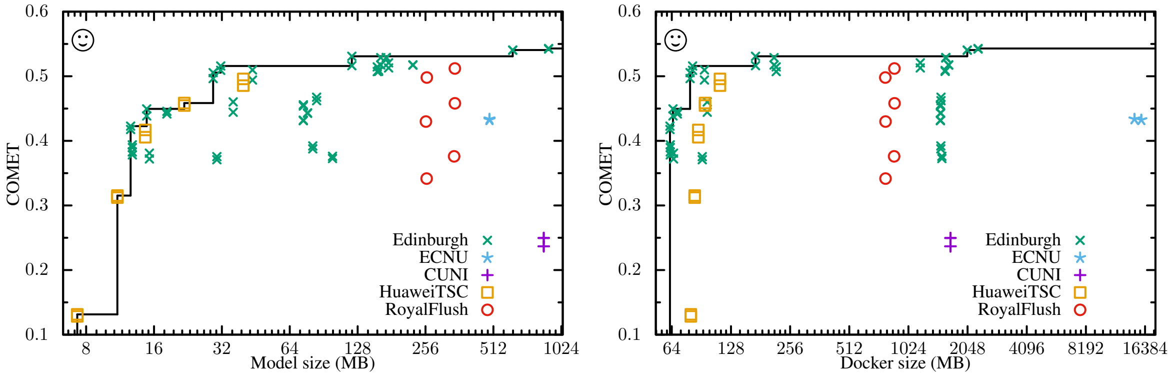


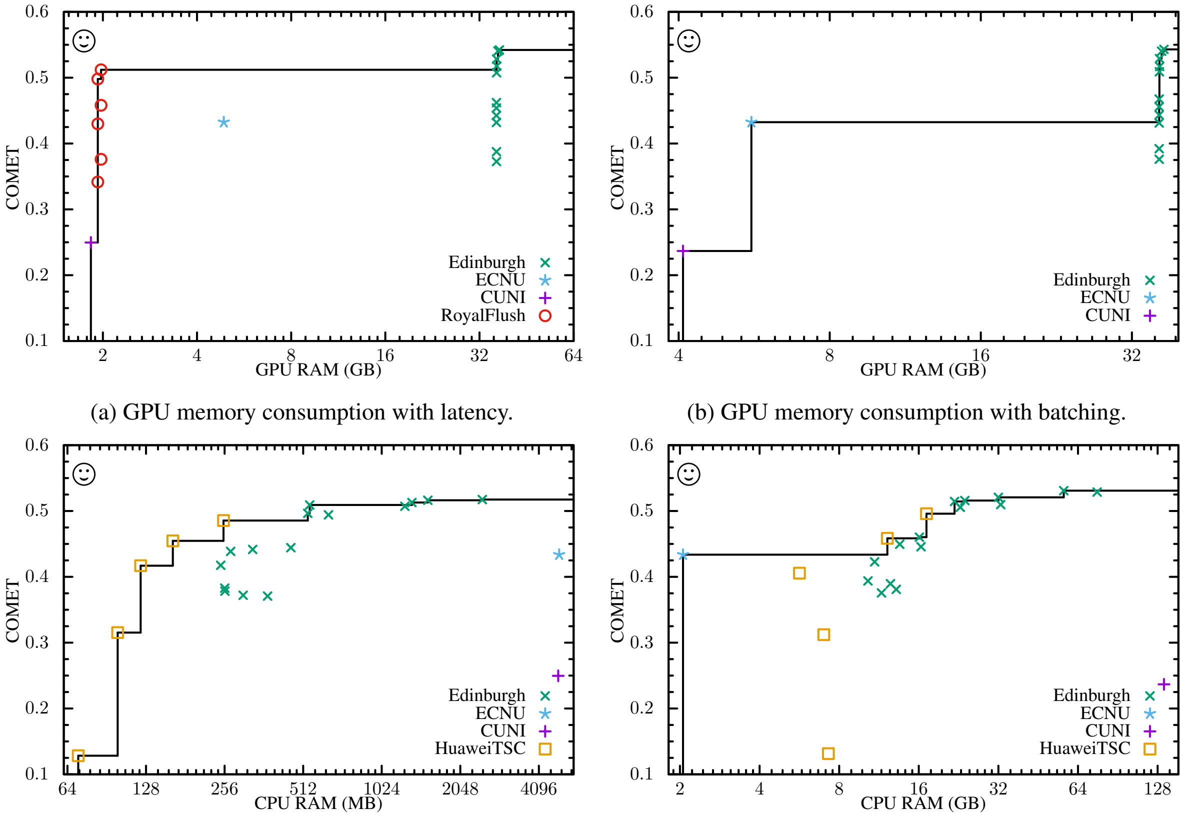
Figure 3: Measured latency for CPU and GPU systems with COMET scores.



(a) Compressed model size.

(b) Compressed Docker image size.

Figure 4: COMET score of systems as a function of model size, and Docker image size. Sizes are reported after compression with xz, and are shown on a logarithmic scale. Some participants did not seek to prune image size and included large Linux installations.



(a) GPU memory consumption with latency.

(b) GPU memory consumption with batching.

(c) 1 Core CPU memory consumption with latency.

(d) 36 Core CPU memory consumption with batching.

Figure 5: RAM consumption of all submissions on a logarithmic scale. Some participants used large batches to favor speed over memory consumption.

seconds on an A100 GPU that costs $3.05/hr in a cloud. That is $0.002/million characters. By comparison, Google Translate's cost is $20/million characters.[1]

In terms of translation throughput cost per $ spent, the GPU submissions are better value for money, provided that enough sentences can be fed to the GPU continuously.

The GPU latency track had been intended to attract non-autoregressive machine translation submissions in their ideal condition with a large GPU and no batch to parallelize. For the first time this year, we had a mix of autoregressive, semi-autoregressive and non-autoregressive systems:

- CUNI submitted a fully non-autoregressive system based on connectionist-temporal-classification (CTC) networks (Helcl et al., 2022).

- Edinburgh submitted bidirectional decoder based semi-autoregressive system (Zhang et al., 2020). This system generates two tokens at an autoregressive step at a time from both sides of the sentences.

- RoyalFlush submitted a semi-autoregressive system based on their novel hybrid regressive translation framework (HRT). They first perform a coarse-grained autoregressive pass that generates some words in the target sentence, with gaps of up to several words in between. Afterwards a second, non-autoregressive pass fills in all the missing words.

The RoyalFlush system proves extremely well suited to the GPU latency task, dominating the pareto frontier in the lower quality setting, even outperforming CPU systems, which have traditionally won this task.

Finally, we note that in semi-autoregressive models and non-autoregressive models, a small drop in BLEU results in a large drop in COMET compared to an autoregressive system, as evidenced by all teams who submitted any form of non-autoregressive MT to the task. This corroborates the findings of (Helcl et al., 2022) where the large discrepancies between BLEU and COMET were noted. We urge participants in future editions of the task to examine manually the output of their non-autoregressive systems.

7 Future tasks

This year's shared task had an increased number of participants, likely due to the organisers providing the distilled data and therefore substantially decreasing the computational cost to participants. We intend to keep this format of the task for future years, in the hopes of attracting even more participants.

German is a high-resource language, which raises the computational cost of participation. We would be interested in also potentially including a medium resource language for distillation so that we can see if the methods that work on high-resource languages generalize well to lower-resource languages, or languages with more morphological complexity.

Last year (Heafield et al., 2021) the organisers suggested that an efficient training shared task would be an interesting natural extension to the efficient translation shared task, however it has proven difficult to set up in practice: we are conscious that the validity of such a task can be easily undermined by participants finding a favorable random seed that fits the training data, or more egregiously by including evaluation data in their training data. We are looking for potential solutions to these problems and we are open to suggestions for next year's edition of the task.

Acknowledgements

The computations described in this research were performed using the Baskerville Tier 2 HPC service (https://www.baskerville.ac.uk/). Baskerville was funded by the EPSRC and UKRI through the World Class Labs scheme (EP/T022221/1) and the Digital Research Infrastructure programme (EP/W032244/1) and is operated by Advanced Research Computing at the University of Birmingham.

Intel Corporation has supported organization. Cloud credits were provided by the Oracle for Research program.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias

[1] https://cloud.google.com/translate/pricing

Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Alexandra Birch, Andrew Finch, Minh-Thang Luong, Graham Neubig, and Yusuke Oda. 2018. Findings of the second workshop on neural machine translation and generation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 1–10, Melbourne, Australia. Association for Computational Linguistics.

Ondřej Dušek, Jan Hajič, Jaroslava Hlaváčová, Jindřich Libovický, Pavel Pecina, Aleš Tamchyna, and Zdeňka Urešová. 2017. Khresmoi summary translation test data 2.0. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Hiroaki Hayashi, Yusuke Oda, Alexandra Birch, Ioannis Konstas, Andrew Finch, Minh-Thang Luong, Graham Neubig, and Katsuhito Sudoh. 2019. Findings of the third workshop on neural generation and translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 1–14, Hong Kong. Association for Computational Linguistics.

Kenneth Heafield, Hiroaki Hayashi, Yusuke Oda, Ioannis Konstas, Andrew Finch, Graham Neubig, Xian Li, and Alexandra Birch. 2020. Findings of the fourth workshop on neural generation and translation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 1–9, Online. Association for Computational Linguistics.

Kenneth Heafield, Qianqian Zhu, and Roman Grundkiewicz. 2021. Findings of the WMT 2021 shared task on efficient translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 639–651, Online. Association for Computational Linguistics.

Jindřich Helcl, Barry Haddow, and Alexandra Birch. 2022. Non-autoregressive machine translation: It's not as fast as it seems. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1780–1790, Seattle, United States. Association for Computational Linguistics.

Niehues Jan, Roldano Cattoni, Stuker Sebastian, Matteo Negri, Marco Turchi, Salesky Elizabeth, Sanabria Ramon, Barrault Loic, Specia Lucia, and Marcello Federico. 2019. The IWSLT 2019 evaluation campaign. In *16th International Workshop on Spoken Language Translation 2019*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Adithya Renduchintala, Denise Diaz, Kenneth Heafield, Xian Li, and Mona Diab. 2021. Gender bias amplification during speed-quality optimization in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 99–109, Online. Association for Computational Linguistics.

Roberts Rozis and Raivis Skadiņš. 2017. Tilde MODEL - multilingual open data for EU languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 263–265, Gothenburg, Sweden. Association for Computational Linguistics.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy. Association for Computational Linguistics.

Biao Zhang, Ivan Titov, and Rico Sennrich. 2020. Fast interleaved bidirectional sequence generation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 503–515, Online. Association for Computational Linguistics.

Findings of the WMT 2022 Shared Task on Automatic Post-Editing

Pushpak Bhattacharyya
IIT Bombay

Rajen Chatterjee
Apple Inc.

Markus Freitag
Google

Diptesh Kanojia
University of Surrey

Matteo Negri
Fondazione Bruno Kessler

Marco Turchi
Zoom Video Communications

Abstract

We present the results from the 8^{th} round of the WMT shared task on MT Automatic Post-Editing, which consists in automatically correcting the output of a "black-box" machine translation system by learning from human corrections. This year, the task focused on a new language pair (English→Marathi) and on data coming from multiple domains (healthcare, tourism, and general/news). Although according to several indicators this round was of medium-high difficulty compared to the past, the best submission from the three participating teams managed to significantly improve (with an error reduction of 3.49 TER points) the original translations produced by a generic neural MT system.

1 Introduction

This paper presents the results of the 8^{th} round of the WMT task on MT Automatic Post-Editing (APE). The task consists in automatically correcting the output of a "black-box" machine translation system by learning from human-revised machine-translated output supplied as training material. The overall task formulation (see Section 2) remained the same as in all previous rounds, where the challenge consisted in fixing the errors present in English documents automatically translated by state-of-the-art, not domain-adapted neural MT (NMT) systems unknown to participants. However, two main factors of novelty characterized the APE 2022 evaluation setting:

- **Language Pair:** This year, we focus on English→Marathi. Marathi is an Indo-Aryan language predominantly spoken by Marathi people in the Indian state of Maharashtra (see Section 3).
- **Data Domain:** Instead of covering one single domain as in previous rounds (either news, medical, or information technology of Wikipedia documents), training/dev/test data were selected from a mix of domains, namely: healthcare, tourism, and general/news.

This year, we had three teams submitting a total of five systems for final evaluation (see Section 5). While the difficulty (Section 4) of this round falls in a medium-high range attested by relatively high baseline results on the test data (20.28 TER / 67.55 BLEU), final results indicate the overall good quality of the submitted runs. Two teams were indeed able to significantly improve over the baseline in terms of the official automatic evaluation metrics (Section 6). In particular, according to the primary metric (*i.e.,* the TER score computed between automatic and human post-edits), the top-ranked system (16.79 TER / 72.92 BLEU) achieved an error reduction of 3.49 TER points. Also, this year, the standard automatic evaluation was complemented by a human evaluation based on direct assessment. However, some problems in the procedure[1] were later discovered, which make it unreliable to draw insights except for the confirmation that two of the three submitted systems were able to improve over the baseline significantly. Specifically, both of them achieved a mean direct assessment score that drastically reduces the gap between the baseline and human post-editing quality. However, due to the mentioned problems in the human evaluation procedure, further details about it will not be included in the discussion below.

Although the different language/domain testing conditions prevent from drawing precise conclusions about the progress of APE technology with respect to last year, the overall positive results confirm its viability for downstream improvements of "black-box" MT systems whose inner workings are not accessible.

[1] Basically, due to an error in assigning the direct assessment tasks, the scores collected can be used to compare systems to the baseline but cannot be used to compare them to each other.

2 Task Description

MT Automatic Post-Editing (APE) is the task of automatically correcting errors in a machine-translated text. As pointed out by (Chatterjee et al., 2015), from the application point of view, the task is motivated by its possible uses to:

- Improve MT output by exploiting information unavailable to the decoder, or by performing deeper text analysis that is too expensive at the decoding stage;
- Cope with systematic errors of an MT system whose decoding process is not accessible;
- Provide professional translators with improved MT output quality to reduce (human) post-editing effort;
- Adapt the output of a general-purpose MT system to the lexicon/style requested in a specific application domain.

This 8^{th} round of the WMT APE shared task kept the same overall evaluation setting of the previous seven rounds. Specifically, the participating systems had to automatically correct the output of an unknown "black-box" MT system (a generic NMT system not adapted to the target domain) by learning from training data containing human revisions of translations produced by the same system. The selected language pair and the data domain, however, were totally new to the task. Different from previous rounds covering more language pairs (or directions), this year focused only on English-Marathi, presenting participants with the traditional source language and, for the third time in a row, an Eastern language as the target. Moreover, while the training, development and test data released in previous rounds were always drawn from a single domain, this year, they covered three domains: healthcare, tourism, and general/news.

3 Data, Metrics, Baseline

3.1 Data

In this round of the APE task, we introduce a new language pair - English-Marathi. Marathi is one of the most spoken Indian languages, with approximately 83 million native speakers and 16 million speakers as a second/third language[2]. Marathi is a known agglutinative language and presents various challenges to machine translation when compared to its other Indian counterparts (Khatri et al., 2021; Banerjee et al., 2021). Moreover, the English-Marathi language pair is considered a low-resource language pair compared to English-Hindi/Bengali/Malayalam (Ramesh et al., 2022) despite having more native speakers around the world. An automatic post-editing approach which helps correct the issues posed by NMT systems is crucial for a low-resource language such as Marathi.

As in all previous rounds, participants were provided with **training** and **development** data consisting of (*source*, *target*, *human post-edit*) triplets. This year, the two sets respectively comprise 18,000 and 1,000 instances, in which:

- The source (SRC) is an English sentence;
- The target (TGT) is a Marathi translation of the source produced by a generic, black-box NMT system unknown to participants. This multilingual NMT system (Ramesh et al., 2022) is based on the Transformer architecture (Vaswani et al., 2017) and is trained on a total of 49 million sentence pairs where the En-Mr parallel corpus is 4.5 million sentence pairs. This parallel data is generic and covers many domains, including the three domains covered by the evaluation setting of this year: healthcare, tourism/culture and general/news.
- The human post-edit (PE) is a manually-revised version of the target, which was produced by native Marathi speakers.

Also this year, a corpus of artificially-generated data has been released as additional training material. It consists of 2 million triplets derived from the *Anuvaad* en-mr parallel corpus[3]. The *Anuvaad* parallel corpus consists of data for 12 language pairs en-X, where X is 12 Indian languages, including Marathi. The English-Marathi data consists of 2.5 million parallel sentences. Specifically, the *source*, *target*, *post-edit* instances of this synthetic corpus are respectively obtained by combining: *i)* the original English source sentence from the *Anuvaad* corpus, *ii)* its automatic translation in Marathi[4], *iii)* the original Marathi target sentence from the *Anuvaad* corpus.

[2] Ethnologue-2022 - Ethnologue has been an active research project since 1951 which maintains online archives of recognized languages list, and their statistics.

[3] https://github.com/project-anuvaad/anuvaad-parallel-corpus

[4] from IndicTrans En-X Model (Ramesh et al., 2022)

Test data consisted of 1,000 (*source*, *target*) pairs, similar in nature to the corresponding elements in the train/dev sets (*i.e.,* same domains, same NMT system). The human post-edits of the target elements were left apart to measure APE systems' performance both with automatic metrics (TER, BLEU) and via manual assessments.

3.2 Metrics

In line with the previous rounds, also this year the plan was to evaluate the participating systems both by means of automatic metrics and, manually, via source-based direct human assessment (Graham et al., 2013). However, as discussed in Section 1, some issues in the manual evaluation procedure were later discovered. For this reason, the discussion of the evaluation results in Section 6 will only concentrate on the automatic metrics. Automatic evaluation was carried out after tokenizing the data using sacremoses[5] and then computing the distance between the automatic post-edits produced by each system for the target elements of the test set, and the human corrections of the same test items. Case-sensitive TER (Snover et al., 2006) and BLEU (Papineni et al., 2002) were respectively used as primary and secondary evaluation metrics. The official systems' ranking is hence based on the average TER calculated on the test set by using the TERcom[6] software: lower average TER scores correspond to higher ranks. BLEU was computed using the multi-bleu.perl package[7] available in MOSES. Automatic evaluation results are presented in Section 6.1.

3.3 Baseline

Also this year, the official baseline results were the TER and BLEU scores calculated by comparing the raw MT output with human post-edits. This corresponds to the score achieved by a "*do-nothing*" APE system that leaves all the test targets unmodified. For each submitted run, the statistical significance of performance differences with respect to the baseline was calculated with the bootstrap test (Koehn, 2004).

4 Complexity Indicators

To get an idea of the difficulty of the task, in previous rounds, we focused on three aspects of the released data, which provided us with information about the possibility of learning useful correction patterns during training and successfully applying them at test time. These are: *i)* repetition rate, *ii)* MT quality, and *iii)* TER distribution in the test set. For the sake of comparison across the eight rounds of the APE task (2015–2022), Table 1 reports, for each dataset, information about the first two aspects. The third one, instead, will be discussed by referring to Figure 1.

4.1 Repetition Rate

The repetition rate (RR), measures the repetitiveness inside a text by looking at the rate of non-singleton n-gram types (n=1...4) and combining them using the geometric mean. Larger values indicate a higher text repetitiveness that may suggest a higher chance of learning from the training set correction patterns that are also applicable to the test set. However, over the years, the influence of repetition rate in the data on system performance was found to be marginal.[8]

Looking at the data released this year, the very low RR values (*i.e.,* 1.46, 0.89, and 0.72 respectively for the SRC, TGT and PE elements) seem to confirm that repetition rate is a scarcely reliable complexity indicator. On one side, these values are close to those observed in rounds were the top-ranked submissions achieved both very large (2020) and very small (2021) gains over the baseline. On the other side, the best result for this year is close to the best results obtained, in previous rounds, on data featuring considerably higher repetition rates (2016, 2017). This suggests that other complexity factors may provide more reliable insights about the difficulty of the task, possibly with an additive effect, still to be fully understood, given by repetition rate.

4.2 MT Quality

Another possible complexity indicator is MT quality, that is the initial quality of the machine-translated (TGT) texts to be corrected. We measure it by computing, the TER (↓) and BLEU (↑) scores (Basel. TER/BLEU rows in Table 1) using the human post-edits as reference. In principle, higher quality of the original translations leaves the APE systems with smaller room for improvement since they have, at the same time, less to learn during

[5] https://pypi.org/project/sacremoses/

[6] http://www.cs.umd.edu/~snover/tercom/

[7] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

[8] The analyses carried out over the years produced mixed outcomes, with impressive final results obtained in spite of low repetition rates (Chatterjee et al., 2020) and vice-versa (Chatterjee et al., 2018, 2019; Akhbardeh et al., 2021).

	Lang.	Domain	MT type	RR_SRC	RR_TGT	RR_PE	Basel. BLEU	Basel. TER	δ TER
2015	en-es	News	PBSMT	2.9	3.31	3.08	n/a	23.84	+0.31
2016	en-de	IT	PBSMT	6.62	8.84	8.24	62.11	24.76	-3.24
2017	en-de	IT	PBSMT	7.22	9.53	8.95	62.49	24.48	-4.88
2017	de-en	Medical	PBSMT	5.22	6.84	6.29	79.54	15.55	-0.26
2018	en-de	IT	PBSMT	7.14	9.47	8.93	62.99	24.24	-6.24
2018	en-de	IT	NMT	7.11	9.44	8.94	74.73	16.84	-0.38
2019	en-de	IT	NMT	7.11	9.44	8.94	74.73	16.84	-0.78
2019	en-ru	IT	NMT	18.25	14.78	13.24	76.20	16.16	+0.43
2020	en-de	Wiki	NMT	0.65	0.82	0.66	50.21	31.56	-11.35
2020	en-zh	Wiki	NMT	0.81	1.27	1.2	23.12	59.49	-12.13
2021	en-de	Wiki	NMT	0.73	0.78	0.76	71.07	18.05	-0.77
2022	en-mr	healthcare/ tourism/news	NMT	1.46	0.89	0.72	67.55	20.28	-3.49

Table 1: Basic information about the APE shared task data released since 2015: languages, domain, type of MT technology, repetition rate and initial translation quality (TER/BLEU of TGT). The last column (δ TER) indicates, for each evaluation round, the difference in TER between the baseline (*i.e.,* the "*do-nothing*" system) and the top-ranked submission.

training and less to correct at the test stage. On one side, training on good (or near-perfect) automatic translations can drastically reduce the number of learned correction patterns. On the other side, testing on similarly good translations can *i)* drastically reduce the number of corrections required and the applicability of the learned patterns, and *ii)* increase the chance of introducing errors, especially when post-editing near-perfect TGTs. The findings of all previous rounds of the task support this observation, which is corroborated by the high correlation (>0.83) between the initial MT quality ("Basel. TER" in Table 1) and the TER difference between the baseline and the top-ranked submission ("δ TER" in Table 1).

As discussed in Section 6, this year seems to confirm the trends observed in the past, albeit with a less evident match. The quality of the initial translations (20.28 TER / 67.55 BLEU) places this round among those of medium-high difficulty (20.0<TER<25.0) for which, except in one case (2015[9]), the performance gains obtained by the top-ranked submissions fall in the range -3.2<δ TER<-6.2. The δ TER of this year (-3.49) also falls in this range, confirming the correlation between the quality of the initial translations and the actual potential of APE.

4.3 TER Distribution

A third complexity indicator is the TER distribution (computed against human references) for the translations present in the test sets. Although TER distribution and MT quality can be seen as two sides of the same coin, it's worth remarking that, even at the same level of overall quality, more/less peaked distributions can result in very different testing conditions. Indeed, as shown by previous analyses, harder rounds of the task were typically characterized by TER distributions particularly skewed towards low values (*i.e.,* a larger percentage of test items having a TER between 0 and 10). On one side, the higher the proportion of (near-)perfect test instances requiring few edits or no corrections at all, the higher the probability that APE systems will perform unnecessary corrections penalized by automatic evaluation metrics. On the other side, less skewed distributions can be expected to be easier to handle as they give automatic systems larger room for improvement (*i.e.,* more test items requiring - at least minimal - revision). In the lack of more focused analyses on this aspect, we can hypothesize that in ideal conditions from the APE standpoint,

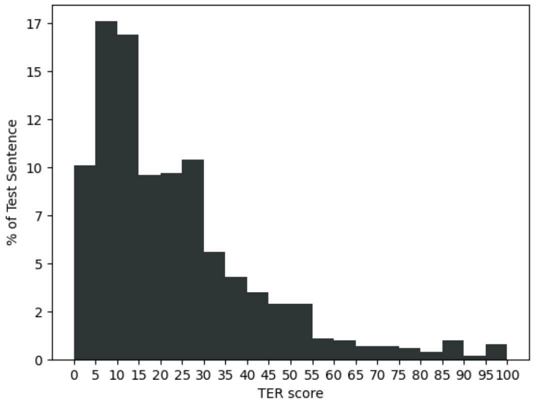


Figure 1: TER distribution in the APE 2022 English-Marathi test set.

[9] The 2015 round is the one in which the APE task was launched. It is somehow an exception being one of the two cases in which none of the participants managed to beat the *do-nothing* baseline (the other one was the 2019 sub-task on English-Russian, also exceptional in the choice of the target language).

ID	Participating team
IITB	Computation for Indian Language Technology - IIT Bombay, India (Deoghare and Bhattacharyya, 2022)
IIIT-Lucknow	IDIAP Research Institute, Switzerland
LUL	Samsung Research and Communication University of China, China (Xiaoying et al., 2022)

Table 2: Participants in the WMT22 Automatic Post-Editing task.

the peak of the distribution would be observed for "post-editable" translations containing enough errors that leave some margin for focused corrections but not too many errors to be so unintelligible to require a whole re-translation from scratch.[10]

Also, with respect to this complexity indicator, the APE 2022 test set can be considered of medium-high difficulty compared to the past rounds. As shown in Figure 1, the TER distribution is quite skewed towards lower values (about 45% of the samples fall in the 15<TER<45 interval) but only 10% of the items can be considered as perfect or near-perfect translations (*i.e.,* 0<TER<5). These values are lower compared to those observed in the test data of harder rounds and higher compared to those observed in the test data of easier rounds.[11] All in all, the improvements over the baseline observed this year for two of the three participating systems (respectively -3.49 and -1.22 TER for the top-ranked and the second-best one) seem to confirm the correlation between TER distribution and task difficulty. However, weighing and understanding the actual contribution of TER distribution and MT quality, together with the possible additive effect of RR, remains a topic for more focused future research.

5 Submissions

As shown in Table 2, this year we received submissions from three teams. Two of them (IIIT-Lucknow and LUL) submitted two runs, while the third one (IITB) participated with only one submission. The main characteristics of two of the three participating systems are summarized below.[12]

Samsung Research and Communication University of China (LUL). This team participated with a Transformer-based system built using fairseq (Ott et al., 2019). Their submissions are characterized by two main aspects: data augmentation and the use of a mixture of experts' approach (Jacobs et al., 1991). Data augmentation is pursued by generating synthetic triplets by means of both an in-house MT system and an external system (Google Translate). The former is used to translate text drawn from several resources, while the latter is used to back-translate the post-edits in the APE training set. The resulting material is combined in different ways so as to obtain different data sets for model fine-tuning. The mixture of experts' approach exploits three domain-specific adapters (Bapna and Firat, 2019; Pham et al., 2020), which are added to the decoder of the base APE model. At inference time, a classifier (added after the encoder) is used to decide which adapter has to be activated.

Computation for Indian Language Technology - IIT Bombay (IITB). This team participated with a Transformer-based system. It exploits a multi-source approach similar to the one in (Chatterjee et al., 2017), with two separate encoders to generate representations for SRC, MT and one decoder. The model is trained with a curriculum learning strategy similar to the one applied by the 2021 winning system (Oh et al., 2021). This is done by first incrementally using out-/in-domain synthetic data (*i.e.,* those released to participants and additional

[10] For instance, based on the empirical findings reported in (Turchi et al., 2013), TER=0.4 is the threshold that, for human post-editors, separates the "post-editable" translations from those that require complete rewriting from scratch.

[11] Although the final results are not comparable due to the different evaluation settings (*i.e.,* different target languages and data domains), the findings from the last two rounds of the APE task provide good examples. In the 2021 round (English-German), where the top submission achieved a small TER reduction compared to the baseline (-0.77), more than 35% of the test instances featured a TER between 0 and 5 and almost 50% of them had 0<TER<10. In contrast, in the 2020 round (English-Chinese) where the top submission achieved the largest baseline improvement ever observed (-12.13), less than 1% of the test samples had 0<TER<5 and ∼89% of them had 40<TER<85.

[12] The IIIT-Lucknow did not produce a system description paper and is left out of our analysis.

ones generated via MT) and then by fine-tuning the model on the real APE data. To ensure the quality of the training material, the LaBSE technique (Language-agnostic BERT sentence embedding) by Feng et al. (2022) is used to filter out low-quality synthetic triplets. To reduce over-correction, a sentence-level quality estimation system trained on the WMT-22 QE English-Marathi sub-task is used to select the final output between an original translation and the corresponding (corrected) version generated by the APE model.

6 Results

6.1 Automatic Evaluation

Participants' results are shown in Table 3. The submitted runs are ranked based on the average TER (case-sensitive) computed using human post-edits of the MT segments as a reference, which is the APE task's primary evaluation metric. We also report the BLEU score, computed using the same references, which represents our secondary evaluation metric.

As it can be seen from the table, the two rankings are coherent: the top submission (16.79 TER, 72.92 BLEU) is the same, and the top three systems outperform by a large margin ($\sim$1 TER and $\sim$2 BLEU scores) the *do nothing* baseline, both in term of BLEU and TER score. These systems are statistically better than the baseline. This is indeed an interesting result showing the effectiveness of the APE systems and confirming their capability of profitably leveraging additional and external resources compared to the MT system.

Looking at relationships between the primary and contrastive submissions (IIT and LUL), the contrastive system shows slightly better performance of the primary submission in one case (LUL). This highlights the difficulty to select the best configuration during system development and indirectly confirms the difficulty to handle APE data characterized by high MT quality, and TER distribution skewed towards perfect/near-perfect translations.

6.2 Systems' Behaviour

Modified, improved and deteriorated sentences. To better understand the behaviour of each APE system, we now turn an eye toward the changes made by each system to the test instances. To this aim, Table 4 shows, for each submitted run, the number of modified, improved and deteriorated sentences, as well as the overall system's precision (*i.e.,* the proportion of improved sentences out of the total number of modified instances for which improvement/deterioration is observed). It's worth noting that, as in the previous rounds, the number of sentences modified by each system is higher than the sum of the improved and the deteriorated ones. This difference is represented by modified sentences for which the corrections do not yield any TER variations. This grey area, for which quality improvement/degradation can not be automatically assessed, would contribute to motivating the integration of human assessments, as done previously.

As it can be seen from the table and similarly to last year's edition, the top systems have been quite conservative in applying their edits by modifying a limited percentage of sentences ($\sim$50% on average, 45.2 for the top submission). Considering the TER distribution where a large number of samples lay in the 15<TER<45 interval, there is the possibility of substantially changing the MT outputs to achieve better performance. This limited number of edits is unexpected and similar to more difficult test sets with more skewed TER distributions toward near-perfect translations. However, systems' final scores are inversely proportional to their aggressiveness showing that limiting the APE edits and carefully selecting them is the right strategy toward significant improvements in quality.

Precision-wise, this year's systems reached 63.9 (in 2021 it was 51.12 and 58.0 in 2020) on average with the best run peaking at 69.49 (vs 53.96 in 2021 and 69.0 in 2020). It is important to note that the average value is significantly affected by the low-performing systems having a precision close to 0. Looking at the percentage of improved (55.6 on average, 63.49 for the top submission) and deteriorated (31.2 on average, 27.87 for the winning system) sentences, the results confirm the capability of the top systems to minimize the wrong changes. Compared to the last editions, the percentage of the improved sentences is among the largest ones achieved by the all-time submitted APE systems.

Edit operations. Similar to previous rounds, we analysed systems' behaviour also in terms of the distribution of edit operations (insertions, deletions, substitutions and shifts) done by each system. This fine-grained analysis of how systems corrected the test set instances is obtained by computing the TER between the original MT output and the output of each primary submission taken as a reference. Sim-

		TER	BLEU
en-mr	IITB_APE_QE_combined_PRIMARY.tsv	**16.79**	**72.92**
	LUL_HyperAug_Adaptor_CONTRASTIVE	**19.06**	**69.96**
	LUL_HyperAug_Finetune_PRIMARY	**19.36**	**69.66**
	baseline (MT)	20.28	67.55
	IIIT-Lucknow_adversia-machine-translation_PRIMARY.txt	57.14	23.43
	IIIT-Lucknow_adversia-machine-translation_CONTRASTIVE.txt	99.81	3.16

Table 3: Results for the WMT22 APE English-Marathi shared task – average TER (↓), BLEU score (↑) Statistically significant improvements over the baseline are marked in **bold**.

Systems	Modified	Improved	Deteriorated	Prec.
IITB_APE_QE_combined_PRIMARY	452 (45.2%)	287 (63.49%)	126 (27.87%)	69.49
LUL_HyperAug_Adaptor_CONTRASTIVE	491 (49.1%)	261 (53.15%)	150 (30.54%)	63.5
LUL_HyperAug_Finetune_PRIMARY	537 (53.7%)	269 (50.09%)	189 (35.19%)	58.73
IIIT-Lucknow_adversia-machine-translation_PRIMARY	999 (99.9%)	46 (0.46%)	929 (92.99%)	0.47
IIIT-Lucknow_adversia-machine-translation_CONTRAS.	1000 (100%)	9 (0.09%)	987 (98.7%)	0.09
Average	69.6 (49.3)	31.4 (55.6)	57.0 (31.2)	38.4 (63.9)

Table 4: Number (raw and proportion) of test sentences modified, improved and deteriorated by each run submitted to the APE 2022 English-Marathi sub-task. The "Prec." column shows systems' precision as the ratio between the number of improved sentences and the number of modified instances for which improvement/deterioration is observed (*i.e.,* Improved + Deteriorated).

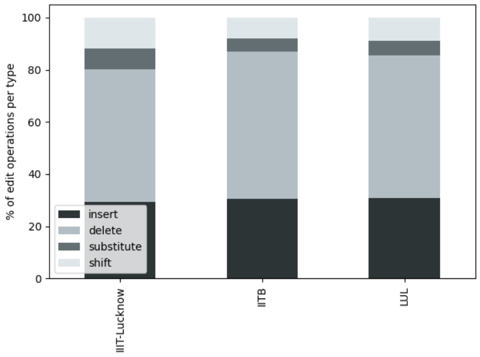


Figure 2: Distribution of edit operations (insertions, deletions, substitutions and shifts) performed by the three primary submissions to the WMT22 APE English-Marathi shared task.

ilar to last year, differences in systems' behaviour are minimal. All of them are characterised by a large number of deletions (∼55.0% on average), followed by insertions (∼30%), shifts (∼10%) and substitutions (∼6%). The system that seems to have a slightly different distribution is IIT-Lucknow resulting in more shifts and substitutions, but these differences are barely visible. Although this year's test set turned out to be simpler than last year (less shewed TER distribution and higher TER), the edit operations are very similar to last year's with a small difference in the number of deletions (65% last year, 55% this year) and insertions (19.2% vs 30%). These variations may depend on the new data, target language and MT system. More thorough future investigations would be needed to find clear explanations for these observations.

7 Conclusion

The 8^{th} round of the shared task on Automatic Post-Editing at WMT was characterized by two main factors of novelty: the language pair (English-Marathi) and the domain of the released data (a mix covering healthcare, tourism, and general/news). Apart from this, the overall setting was the same as in previous recent rounds, in which participating systems had to automatically correct the output of a generic neural MT system, being evaluated with the TER (primary) and BLEU (secondary) automatic metrics. In continuity with the past, also human evaluation via source-based direct assessment was carried out, but it is not discussed in this report due to its unreliable outcomes. In terms of the three complexity indicators discussed in Section 4 (repetition rate, original MT quality and TER distribution), the difficulty of this round falls in a medium-high range. This is reflected by the performance of the systems submitted by the three participating teams: two of them were indeed able to improve over the *do-nothing* baseline with (statistically significant) error reductions up to -3.49 TER points (+5.37 BLEU). Although these results are not comparable with those from previous years due to the different language/domain testing conditions, the observed improvements in the new language direction confirm the viability of APE for downstream improvements of "black-box" MT systems whose inner workings are not accessible.

Acknowledgments

We would like to thank the translation agencies Techliebe, Shri Samarth Krupa Language Solutions, Zibanka, and Desicrew, who helped post-edit the dataset for the English-Marathi language pair.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Aakash Banerjee, Aditya Jain, Shivam Mhaskar, Sourabh Dattatray Deoghare, Aman Sehgal, and Pushpak Bhattacharyya. 2021. Neural machine translation in low-resource setting: a case study in english-marathi pair. In *Proceedings of the 18th Biennial Machine Translation Summit (Volume 1: Research Track)*, pages 35–47.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Rajen Chatterjee, M. Amin Farajian, Matteo Negri, Marco Turchi, Ankit Srivastava, and Santanu Pal. 2017. Multi-source neural automatic post-editing: Fbk's participation in the wmt 2017 ape shared task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 630–638, Copenhagen, Denmark. Association for Computational Linguistics.

Rajen Chatterjee, Christian Federmann, Matteo Negri, and Marco Turchi. 2019. Findings of the WMT 2019 shared task on automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 11–28, Florence, Italy. Association for Computational Linguistics.

Rajen Chatterjee, Markus Freitag, Matteo Negri, and Marco Turchi. 2020. Findings of the WMT 2020 shared task on automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 646–659, Online. Association for Computational Linguistics.

Rajen Chatterjee, Matteo Negri, Raphael Rubino, and Marco Turchi. 2018. Findings of the WMT 2018 shared task on automatic post-editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 710–725, Belgium, Brussels. Association for Computational Linguistics.

Rajen Chatterjee, Marion Weller, Matteo Negri, and Marco Turchi. 2015. Exploring the Planet of the APEs: a Comparative Study of State-of-the-art Methods for MT Automatic Post-Editing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 156–161, Beijing, China. Association for Computational Linguistics.

Sourabh Deoghare and Pushpak Bhattacharyya. 2022. Iit bombay's wmt22 automatic post-editing shared task submission. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous Measurement Scales in Human Evaluation of Machine Translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87.

Jyotsana Khatri, Rudra Murthy, Tamali Banerjee, and Pushpak Bhattacharyya. 2021. Simple measures of bridging lexical divergence help unsupervised neural machine translation for low-resource languages. *Machine Translation*, 35(4):711–744.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain.

Shinhyeok Oh, Sion Jang, Hu Xu, Shounan An, and Insoo Oh. 2021. Netmarble AI center's WMT21 automatic post-editing shared task submission. In *Proceedings of the Sixth Conference on Machine Translation*, pages 307–314, Online. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Minh Quang Pham, Josep Maria Crego, François Yvon, and Jean Senellart. 2020. A study of residual adapters for multi-domain neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 617–628, Online. Association for Computational Linguistics.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2022. Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, pages 223–231.

Marco Turchi, Matteo Negri, and Marcello Federico. 2013. Coping with the subjectivity of human judgements in MT quality estimation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 240–251, Sofia, Bulgaria. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Huang Xiaoying, Lou Xingrui, Zhang Fan, and Tu Mei. 2022. Lul's wmt22 automatic post-editing shared task submission. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Embarrassingly Easy Document-Level MT Metrics: How to Convert Any Pretrained Metric Into a Document-Level Metric

Giorgos Vernikos[*] EPFL + HEIG-VD
Brian Thompson AWS AI Labs
Prashant Mathur AWS AI Labs
Marcello Federico AWS AI Labs
georgios.vernikos@epfl.ch, {brianjt, pramathu, marcfede}@amazon.com

Abstract

We present a very simple method for extending pretrained machine translation metrics to incorporate document-level context. We apply our method to four popular metrics: BERTScore, Prism, COMET, and the reference-free metric COMET-QE. We evaluate our document-level metrics on the MQM annotations from the WMT 2021 metrics shared task and find that the document-level metrics outperform their sentence-level counterparts in about 85% of the tested conditions, when excluding results on low-quality human references. Additionally, we show that our document-level extension of COMET-QE dramatically improves accuracy on discourse phenomena tasks, supporting our hypothesis that our document-level metrics are resolving ambiguities in the reference sentence by using additional context.

1 Introduction

Automatic evaluation is crucial to the machine translation (MT) community for tracking progress, evaluating new ideas and making modeling choices. While human evaluation is the gold standard for MT evaluation, it is very expensive, and thus most research groups must rely on automatic metrics. Current State-of-the-art (SOTA) metrics are *pretrained* (Kocmi et al., 2021; Freitag et al., 2021b), leveraging existing language models (LMs) or sequence-to-sequence models to judge how well a hypothesis (i.e. MT system output) conveys the same meaning as a human reference translation.

Sentences are often ambiguous, and many recent works have demonstrated that incorporating inter-sentential (i.e. document-level) context is beneficial in both MT (Lopes et al., 2020; Fernandes et al., 2021) and human evaluation of MT (Läubli et al., 2018; Toral, 2020; Freitag et al., 2021a).

A human reference translation is (at least ideally) created taking the entire source document into account. However, just as source sentences are often ambiguous, we hypothesize that human reference sentences also contain ambiguities. Thus, when a system output deviates from the human reference, we may need to look at additional context to determine if those deviations are acceptable, in the context of the full document translation.

In this study, we present a simple procedure for extending pretrained MT metrics to the document level. Prior work has used pretrained models models like BERT (Devlin et al., 2019) to embed a single human reference sentence and hypothesis (e.g. an MT output) sentence. We instead argue that a *better* representation of the reference or hypothesis sentence can be obtained by providing several sentences of context to the pretrained model, allowing the pretrained model to *use surrounding context when embedding each sentence of interest*. Once the embeddings of the reference or hypothesis sentence have been computed (taking into account surrounding sentence context), the metric is computed in the same manner as the sentence-level metric.[1,2]

We apply this method to extend four popular pretrained metrics to the document level:[3]

- BERTScore (Zhang et al., 2020), a text generation metric that uses the alignments from token embeddings of a pretrained BERT model to score the similarity of a hypothesis and reference.
- Prism (Thompson and Post, 2020a), a text generation metric which utilizes a sequence-to-sequence paraphrase model to score how well a hypothesis paraphrases the reference.
- COMET (Rei et al., 2020), an MT metric which fine-tunes a multilingual LM, namely

[*] Work conducted during an internship at Amazon.

[1] In the case of Prism (Thompson and Post, 2020a), we modify this logic slightly to retain only the probabilities of the sentence of interest (see § 3.2).

[2] In the case of COMET/COMET-QE (Rei et al., 2020), which incorporates the source sentence, we provide additional source context in the same manner (see § 3.3 and § 3.4).

[3] We release our code at `https://github.com/amazon-research/doc-mt-metrics`.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 118 - 128
December 7-8, 2022 ©2022 Association for Computational Linguistics

XLM-R (Conneau et al., 2020), to predict translation quality given a hypothesis, source, and reference.

- COMET-QE (Rei et al., 2020), the reference-free (i.e. "quality estimation as a metric") version of COMET.

To test the effectiveness of our document-level metrics, we measure system-level correlation with human judgments. We select the so-called "platinum" Multidimensional Quality Metrics (MQM) judgments collected for the WMT 2021 metrics task (Freitag et al., 2021b). We believe MQM judgments are the best available to test document-level MT metrics as these judgments are made by expert translators that have access to—and are strongly advised to consider—source-side document-level context when judging each target sentence. We perform evaluation on all the WMT 2021 language pairs (En→De, Zh→En, En→Ru) and domains (TED talks and news) for which MQM judgments are available.

We find that our document-level extensions of these four metrics outperform their sentence-level counterparts in 75% of cases considered. Excluding Zh→En news, where the human reference is of low quality (see § 4.1), we see improvements in 85% of cases. This provides strong evidence that document-level context is useful in the automatic evaluation of MT.

We also conduct analysis to better understand the performance improvement that we observe. We demonstrate that our document-level extension of COMET-QE significantly improves over its sentence-level counterpart on targeted tasks evaluating discourse phenomena, namely pronoun resolution and Word Sense Disambiguation (WSD).[4] This finding provides further evidence that our document-level metrics are using context to resolve ambiguities in the reference sentence. We also show that using reference context is better than using context from the MT output, likely because the MT output contains more errors than the reference.

In summary, our contributions are:

1. We present a simple but effective method to extend pretrained sentence-level metris to the document level, and apply it to four popular metrics.
2. We show that the proposed document-level metrics tend to have better correlation with human judgments than their sentence-level counterparts.
3. We improve over both COMET and COMET-QE, which appear to be the previous SOTA automatic metric and reference-free metric, respectively (Freitag et al., 2021b; Kocmi et al., 2021).
4. We conduct analysis to show that the improvements observed using our approach can be attributed to better context utilization, and also show that using reference context is better than using context from the hypothesis.

2 Related Work

Our work has parallels in human MT evaluation, where document-level judgments are required to distinguish human translation quality from MT system quality (Läubli et al., 2018; Toral, 2020). Castilho et al. (2020) showed that many source sentences are ambiguous, but that ambiguities are often resolved using only a few additional sentences of context. This suggests that we do not need to incorporate very many additional sentences of context into a document-level metric in order to see an improvement in quality.

Pretrained metrics are metrics which leverage large existing pretrained LMs or sequence-to-sequence models, and include YiSi (Lo, 2019), COMET (Rei et al., 2020), BERTscore (Zhang et al., 2020), Prism (Thompson and Post, 2020a), BLEURT (Sellam et al., 2020), and others. Pretrained metrics have been shown to consistently outperform surface-level metrics such as BLEU (Papineni et al., 2002), TER (Snover et al., 2006), and chrF (Popović, 2015) – see Mathur et al. (2020); Kocmi et al. (2021); Freitag et al. (2021b).

Prior to the rise of pretrained metrics, several works targeted discourse-level phenomena in MT metrics such as pronominal anaphora (Hardmeier and Federico, 2010; Miculicich Werlen and Popescu-Belis, 2017; Jwalapuram et al., 2019) and lexical cohesion (Wong and Kit, 2012; Gong et al., 2015). For a detailed overview of evaluation of discourse-level phenomena, we direct the reader to Maruf et al. (2021). Recently, Jiang et al. (2022) proposed BlonDe, a document-level metric that focuses on discourse phenomena in order to score a translated document. However, we find that BlonDe substantially under-performs modern pretrained metrics, despite taking advantage of document-level context (see § 5.1).

[4] The use of a reference would make these tasks trivial, so we limit our analysis to the reference-free COMET-QE.

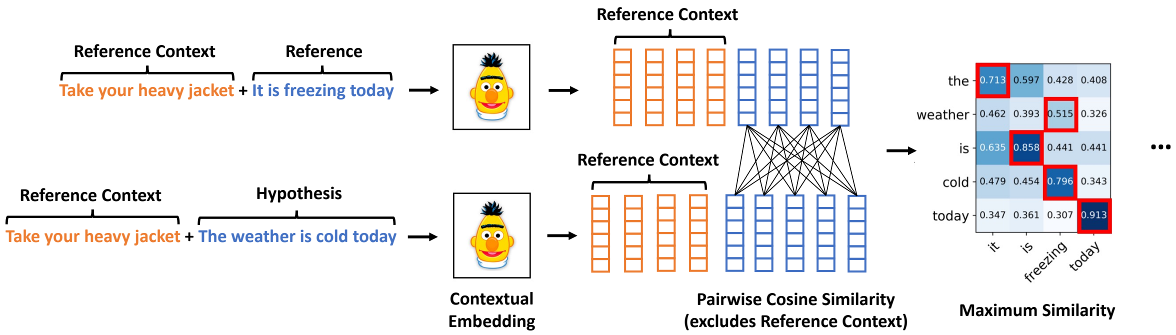

Reference Context
Reference
Take your heavy jacket + It is freezing today
Reference Context
Hypothesis
Take your heavy jacket + The weather is cold today
Contextual Embedding
Reference Context
Reference Context
Pairwise Cosine Similarity (excludes Reference Context)
the 0.713 0.597 0.428 0.408
weather 0.462 0.393 0.515 0.326
is 0.635 0.858 0.441 0.441
cold 0.479 0.454 0.796 0.343
today 0.347 0.361 0.307 0.913
it is freezing today
Maximum Similarity
...

Figure 1: To extend BERTScore to the document level, we add reference context (e.g. "Take your heavy jacket") to both the reference sentence (e.g. "It is freezing today") and hypothesis sentence (e.g. "The weather is cold today"). This context is used to improve the embeddings of the reference and hypothesis sentences (e.g. helping the model understand that "it" is likely referring to weather). However, the additional context is not used when performing alignment and scoring, which follows standard sentence-level BERTScore. The same methodology is applied to Prism and COMET/COMET-QE (not shown). Image adapted from Zhang et al. (2020).

3 Method

At a high level, we propose a very simple procedure for extending pretrained MT metrics to the document level: As in standard sentence-level metrics, we produce a score for a single hypothesis sentence compared to a single human reference translation sentence. However, we use additional context[5,6] from the reference translation when computing the contextual embeddings for both the hypothesis sentence and reference sentence. Once the hypothesis and reference sentence have been embedded, we discard the extra context sentences before computing metric scores following the same process as the corresponding sentence-level metric. Additional details are provided for each metric below.

For the following discussion, let s refer to the source sentence, h refer to the hypothesis (i.e. MT system output) sentence, r refer to the human reference translation sentence, and let c_s, c_h and c_r refer to the source, hypothesis, and reference context, respectively.

3.1 Document-level BERTScore

BERTScore (Zhang et al., 2020) is an unsupervised text generation metric that leverages the power of a pretrained large LM to score generated text. BERTScore encodes tokens of both the reference and the hypothesis with a pretrained LM and computes soft alignments based on token similarities. The alignment matrix is then used to calculate the precision, recall and F1 scores of the hypothesis compared to the reference.

To extend BERTScore to the document level, we use the reference context $\langle c_r \rangle$ while encoding the hypothesis or the reference with the LM. However, we align only the tokens of the reference/hypothesis sentence being scored (see Figure 1 for an illustration).

For BERTScore we use the default LM option for each language pair, which is the multilingual BERT-base (Devlin et al., 2019) for all En→* pairs and RoBERTa-large (Liu et al., 2019) for *→En pairs. BERT and RoBERTa are naively document-level; specifically, the LMs are trained on up to 512 tokens at a time, which is significantly longer than the average sentence length. Thus no changes to the underlying model were required to extend BERTscore to the document level.

3.2 Document-level Prism

Prism (Thompson and Post, 2020a,b) is an unsupervised text generation metric that uses a sequence-to-sequence paraphraser to evaluate how well a hypothesis paraphrases a human reference translation. Specifically, to score a translation the reference is fed to the encoder and the hypothesis is force-decoded in the decoder via teacher forcing. The token-level probabilities of the reference are aggregated to produce a score and the process is repeated with the hypothesis in the encoder side and the reference in the decoder. The final score is the average of the two scores.

In order to generalize Prism for document-level

[5] We use two preceding sentences from the reference as context, but our method could be applied to additional previous and/or subsequent sentences.

[6] We only use valid context. For example, when using a nominal value of two prior sentences as context, the first sentence in a document gets no context sentences and the second sentence gets one context sentence.

evaluation we concatenate the reference context c_r to both the reference and hypothesis $\langle c_r; r, c_r; h\rangle$. The context is used as a prompt; that is, we only aggregate token-level probabilities for the sentence being evaluated. The authors of Prism release the sentence-level multilingual MT model that they zero-shot paraphrase model. However, we require a document-level model to extend Prism to the document level. One option for extending Prism to the document level is to train a document-level, multilingual MT model. While document-level data collection methods and datasets do exist (Guo et al., 2019; Thompson and Koehn, 2020; Cettolo et al., 2012; Lison et al., 2018), document-level data is not currently available in nearly as many language pairs as sentence-level data. To extend Prism to the document level, we instead use mBART-50 (Tang et al., 2020), a multilingual encoder-decoder LM. mBART-50 is trained on document fragments of up to 512 tokens, in 50 languages, resulting in a multilingual document-level paraphraser. Note that while an mBART model fine-tuned on (sentence-level) translations is available, we do not use it because we require a document-level model. As a result, although the mBART model we use is multilingual, it is not a translation model so we cannot use it for the reference-free version of Prism.

3.3 Document-level COMET

COMET (Rei et al., 2020) is a supervised metric that is trained on human judgments. COMET encodes the source, hypothesis and reference via a multilingual pretrained LM and the representation of each sentence is the average of its output token embeddings. The encoded representations are further combined via subtraction and multiplication and fed to a regressor that predicts a score for each translated sentence. We use COMET-MQM_2021 (Rei et al., 2021), which is built on top of XLM-RoBERTa-large (Conneau et al., 2020). The COMET models are pretrained on direct assessment judgements from WMT 2015 to WMT 2020 and fine-tuned on MQM z-scores from Freitag et al. (2021a).

To extend COMET to the document level, we integrate source context c_s and reference context c_r by concatenating them with the source and hypothesis/reference in the encoder. We obtain sentence representations by averaging the output embeddings of the tokens of the current sentence only before passing them to the regressor.

As with BERTscore, the model underlying COMET is inherently document-level. However, the underlying LM is fine-tuned for a few epochs on human judgments from previous WMT campaigns that consist of a single (source, reference, and hypothesis) sentence and the corresponding score. As the amount of fine-tuning is quite limited, we hypothesize that the model has still retained its ability to handle text beyond sentence level, and this assumption appears to be confirmed by experimental results (see § 5.1).

3.4 Document-level COMET-QE

COMET-QE (Rei et al., 2021) is the reference-free version of COMET. We use the latest COMET-MQM-QE_2021, trained similarly to the COMET-MQM_2021 discussed above. Although COMET-QE does not does not have access to the reference it has been shown to perform reasonably well compared to strong reference-based metrics (Kocmi et al., 2021).

Similar to reference-based COMET, to extend COMET-QE to the document level, for each source s and hypothesis h, we concatenate the previous source and hypothesis sentences as context $\langle c_s; s, c_h; h\rangle$ and score the hypothesis h in question.

The pretrained model for COMET-QE is the same as the one used in COMET, therefore no further modifications are required to extend COMET to the document level.

4 Experiments

Motivated by the finding of Scherrer et al. (2019); Kim et al. (2019); Castilho et al. (2020) that two previous sentences are sufficient context to correctly resolve ambiguities in the majority of sentences, we use two previous reference sentences as context unless otherwise noted. Sentences are separated using the separator token of each model: [SEP] for RoBERTa and <\s> for XLM-R and mBART-50. We use reference context c_r as reference for the hypothesis, as opposed to hypothesis context c_h. This is done in order to avoid propagation of translation errors (see § 6.1 for an ablation using hypothesis context instead of reference context).

4.1 Human Judgment Experiments

We compare our document-level metrics judgments of MT outputs with those of the human-generated

Model	Input	TED talks			News		
		En→De	En→Ru	Zh→En	En→De	En→Ru	Zh→En
BlonDe	$\langle c_h, h, c_r, r\rangle$	-	-	-0.232	-	-	0.212
Prism (m39v1)	$\langle h, r\rangle$	0.656	0.867	0.272	0.841	0.799	0.558
Prism (mBART-50)	$\langle h, r\rangle$	0.486	0.845	0.240	0.661	0.710	0.363
Doc-Prism (mBART-50)	$\langle c_r; h, c_r; r\rangle$	**0.692**	**0.852**	**0.372**	**0.825***	**0.777**	**0.374**
BERTScore	$\langle h, r\rangle$	0.506	0.831	0.293	0.930	**0.629**	**0.575***
Doc-BERTScore	$\langle c_r; h, c_r; r\rangle$	**0.613***	**0.836**	**0.344***	**0.948***	0.622	0.535
COMET	$\langle s, h, r\rangle$	**0.818**	0.841	0.266	0.772	0.659	**0.628**
Doc-COMET	$\langle c_s; s, c_r; h, c_r; r\rangle$	0.816	**0.849**	**0.297**	**0.802***	**0.676**	0.513
COMET-QE	$\langle s, h\rangle$	0.694	0.818	**-0.209**	0.711	0.688	**0.529**
Doc-COMET-QE	$\langle c_s; s, c_h; h\rangle$	**0.724**	**0.830**	-0.255	**0.733**	**0.733***	0.462

Table 1: System-level correlation with WMT 2021 MQM annotations for Prism, BERTScore, COMET and COMET-QE and their generalization for document-level evaluation (Doc-*, this work). Within each document/sentence-level pair, **bold** denotes the best correlation and "*" denotes a statistically significant ($p < 0.05$) difference. Excluding Zh→En news data, which has a very low-quality human reference (see § 4.1), our document-level metrics outperform their sentence-level counterparts in 17 of 20 (85%) of cases, and 6 of 6 (100%) of statistically significantly different cases.

MQM annotations from the 2021 WMT metrics shared task (Freitag et al., 2021a). We select MQM for several reasons: They are produced by professional translators (compared to crowd workers or translation researchers) and require explicit error annotations that are believed to lead to higher quality annotations. Also, MQM annotators are specifically instructed to "*identify all errors within each segment in a document, paying particular attention to document context*." In 2021, in addition to the news domain, annotations were also produced for translations of TED talks in three language pairs: En→De, Zh→En and En→Ru.

One potential problem with the metrics dataset is the quality of the Zh→En news human reference. The WMT metrics shared task organizers acquired MQM scores for the human references, in addition to MT system outputs. The Zh→En reference received an MQM score of just 4.27, only slightly better than the best MT system at 4.42 (Freitag et al., 2021b). For reference, 0.0 is a perfect score and a score of 5.0 corresponds to one major error (or many minor errors) per sentence. In contrast, for the same language pair, the TED reference has an MQM score of 0.42 vs the best MT system at 1.65.

4.2 Discourse Phenomena Experiments

In order to confirm that any gains we see from document-level metrics are in fact due to their ability to correctly handle ambiguities in the reference which can be resolved using document-level context, we also perform targeted evaluation of discourse phenomena using contrastive sets. These testsets are common in the evaluation of document-level MT systems where a context-aware model should ideally assign the highest probability to the correct translation; all translations are plausible and only the use of context can reveal the correct translations. For our case, since we are evaluating MT metrics, we treat each sentence as a different hypothesis and calculate how often our metric ranks the correct translation the highest. Since the use of a reference would make this task trivial for reference-based metrics, we only evaluate on COMET-QE. We use ContraPro (Müller et al., 2018), a selection of sentences from OpenSubtitles2018 (Lison et al., 2018) that contain the English anaphoric pronoun *it* in the source side. Starting from the correct translation in German, contrastive translations are automatically created to contain the German pronouns *er*, *sie* and *es*. In order to identify the correct translation the model must look into previous context. We also evaluate on a similar dataset for En→Fr created by Lopes et al. (2020) for the translation of *it* and *they* into *il*, *elle*, *ils*, *elles* in French. Finally, we evaluate on DiscEvalMT (Bawden et al., 2018), a contrastive test which consists of 200 examples of anaphoric pronoun translation for En→Fr and 200 examples of WSD.

4.3 Baseline Methods

For correlation with human MT quality judgments, in addition to the sentence-level version of each metric we extend, we also compare to

Model	En→De			En→Fr				
	Intra	Inter	Total	Intra	Inter	Total	Anaphora	WSD
Lopes et al. (2020)	-	-	70.8	-	-	83.2	82.5	55.0
COMET-QE	78.2	40.9	48.4	76.3	76.6	76.5	50.0	50.0
Doc-COMET-QE (this work)	**80.5**	**72.6**	**74.2**	**88.7**	**88.0**	**88.3**	**83.5**	**68.0**

Table 2: Accuracy (percentage correct) for targeted evaluation of contextual phenomena. Our document-level version of COMET-QE substantially outperforms the sentence-level COMET-QE, and also outperforms the best methods proposed by Lopes et al. (2020), demonstrating that it is successfully incorporating contextual information.

BlonDe (Jiang et al., 2022), an overlap-based document-level metric that focuses on discourse phenomena.[7] We also compare to Prism using the m39v1 model released by the authors of Prism.

For discourse phenomena, we compare our document-level COMET-QE model to the sentence-level COMET-QE as well as the best reported results of Lopes et al. (2020).

5 Results

5.1 Correlation with Human Judgments

We present the system-level Pearson correlation with the human annotations of the 2021 WMT metrics task for all metrics (sentence- and document-level) in Table 1. Statistical significance ($p < 0.05$) is computed for each sentence- vs document-level metric pair following Freitag et al. (2021b) using the PERM-BOTH hypothesis test (Deutsch et al., 2021). We also provide the results of BlonDe (only for *→En since this metric relies on entity taggers and discourse markers that are only trained in English) and Prism with the original model (m39v1) for comparison.

Overall, adding document-level context leads to improved correlation with human judgments for all metrics. Our document-level metrics outperform their sentence-level counterparts in 18 of 24 (75%) of cases considered. Excluding Zh→En news data, which has a very low-quality human reference (see § 4.1), our document-level metrics outperform their sentence-level counterparts in 17 of 20 (85%) of cases. Looking at only pairs with statistically significant differences, our document-level metrics outperform their sentence-level counterparts in 6 of 7 cases (86%), and 6 of 6 (100%) of cases excluding Zh→En news.

We see that document-level metrics outperform sentence-level metrics in only 1 of 4 cases on Zh→En news This suggests that the document-level metrics are sensitive to errors in the reference context. This hypothesis is further supported by analysis in § 6.1.

For Prism, we observe that the sentence-level results with the original m39v1 model are better than the sentence-level results with mBART-50. However, by using document-level context we are able to improve over the sentence-level Prism with mBART-50 in every language pair/domain. This narrows the gap between Prism with mBART and Prism with m39v1, outperforming the stronger m39v1 model in two TED language pairs.

Although the COMET models are fine-tuned on single sentences, experimental results suggest they are able to retain their ability to handle inter-sentential dependencies. We considered retraining COMET excluding older direct assessment judgments which did not take document-level context into account; however this would have severely limited the amount of (already very limited!) training data.

Finally, we observe that BlonDe performs significantly worse than the pretrained metrics as well as our document-level extensions, underperforming everything except document-level COMET-QE in TED Zh→En.

5.2 Discourse Phenomena Improvements

We provide the results of targeted evaluation on contrastive datasets for COMET-QE and Doc-COMET-QE in Table 2. We also provide the scores of the best-performing document-MT model for each dataset from Lopes et al. (2020) for comparison. The reference-based metrics are not considered in this section as the use of a reference would make the task trivial.

We observe that the document-level COMET-QE substantially outperforms the sentence-level COMET-QE, and even outperforms document-

[7] We report BlonDe results in English only, as BlonDe uses a discourse marker script from Sileo et al. (2019) which was trained only in English. BlonDe could likely be extended to other languages but we did not attempt to do so.

	Context	Doc-Prism	Doc-BERTScore	Doc-COMET
hypothesis	$\langle c_s; s, c_r; r, c_h; h\rangle$	0.595	0.624	0.630
reference	$\langle c_s; s, c_r; r, c_r; h\rangle$	**0.649**	**0.650**	**0.659**

Table 3: Average correlation with MQM human judgments of our document-level metrics using previous hypothesis sentences as context vs. previous reference sentence as context. COMET-QE is excluded because it does not depend on the reference. For all three methods, we see better correlation using the reference for hypothesis context. We hypothesize that this is because using previous hypothesis sentences allows for propagation of errors (i.e. an error in a previous sentence can impair the judgment of the current sentence).

level translation models optimized for discourse tasks. Surprisingly, we observe improvements in the evaluation of pronoun translation not only when the necessary information is located in a previous sentence (Inter) but even in the case where the antecedent can be found in the same sentence (Intra), suggesting additional context is helpful in these cases as well. Apart from pronoun translation, our approach also improves over both the sentence-level metric and the document-level MT of Lopes et al. (2020) at WSD. These findings all support our hypothesis that our document-level metrics are resolving ambiguities in the reference sentence by using additional context.

6 Ablations

6.1 Hypothesis vs Reference Context

For our document-level MT metrics described prior to this point, we use the reference context c_r (as opposed to the hypothesis context c_h) as context for the hypothesis. Our reasoning behind this decision is that previous translations could contain errors that might bias the document-level metric into rewarding erroneous translations. To test this, we conduct an ablation experiment in which we concatenate the hypothesis context to the hypothesis while the context of the remaining inputs (i.e. the reference and the source sentence) remains unchanged. Table 3 shows the average correlation across all language pairs and domains using either the hypothesis context or the reference context. We do not provide these scores for COMET-QE as it does not have access to the reference.

We observe that the use of the hypothesis context degrades performance for all metrics, which is in line with the findings of Fernandes et al. (2021) for document-level MT. We suspect that this is because the previous hypothesis sentences contain more errors than previous reference sentences, and thus using previous hypothesis sentences allows for more propagation of errors (i.e. an error in a previous sentence can impair the judgment of the current sentence).

One disadvantage of using reference context for the hypothesis is that we cannot measure document-level fluency, that is, how well a document flows from one sentence to the next. Our analysis suggests that either document level fluency is of less concern than error propagation, and/or that MQM judgments are not adequately capturing document-level fluency.

6.2 Amount of Context

In our experiments so far we have used the previous two sentences as context, motivated by the finding of Scherrer et al. (2019); Kim et al. (2019); Castilho et al. (2020) that two previous sentences are sufficient context to resolve ambiguities in the majority of sentences. Figure 2 shows the results for [0, 1, 2] previous sentences as context for news articles and TED talks. In the news domain we observe that for En→De and En→Ru), adding more context helps. On the other hand, for Zh→En, adding context appears to be harmful. We believe this is likely explained by the relatively low-quality human references in Zh→En (see § 4.1). For TED talks, although the results are somewhat noisy, we also observe that more context tends to improve correlation across all three language pairs.

7 Conclusion

We proposed a simple and effective approach to generalize pretrained MT metrics to the document level. We apply our approach to BERTScore, Prism, COMET-QE, and COMET-QE, and we believe that it could easily be extended to other pretrained sentence-level metrics. To the best of our knowledge, our work is the first example of pretrained document-level MT metrics.

We demonstrate that the use of document-level context in pretrained metrics improves correlation with human judgments, and that the improvements

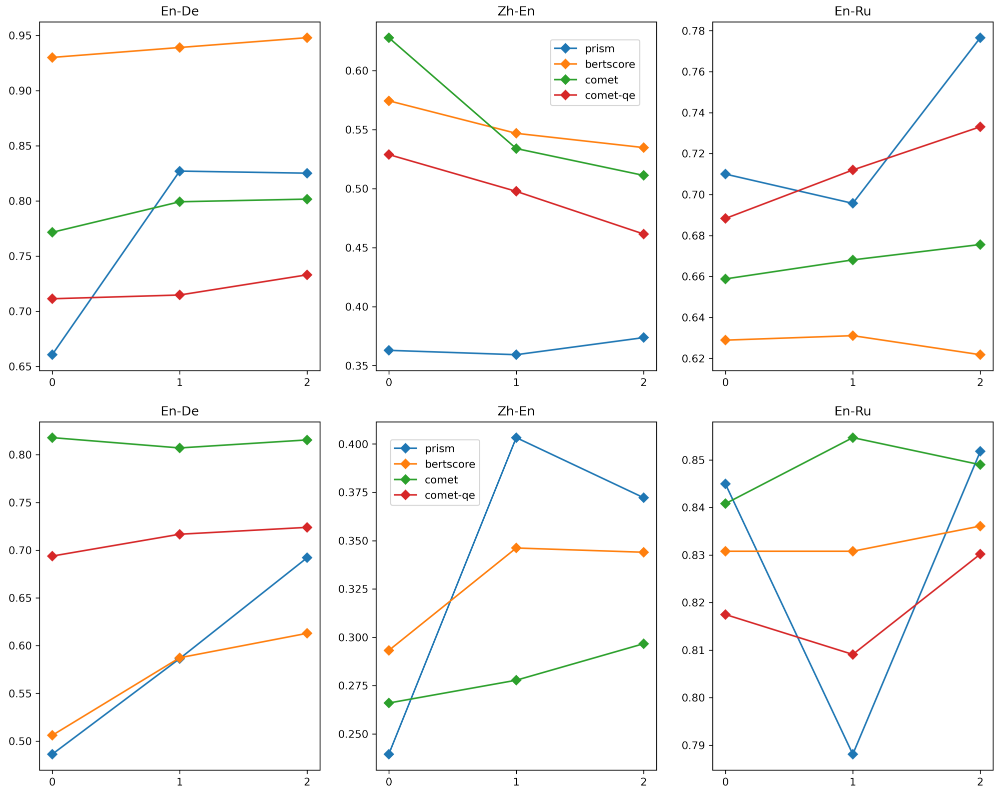


Figure 2: System-level Pearson correlation with human correlation vs. number of sentences of context for News (upper) and TED talks (lower). Although the results are noisy, in general we observe that correlation improves as the amount of context increases. The one exception is Zh→En News, which we attribute to poor human references (see § 4.1).

are likely due to fact that the document-level metrics can resolving ambiguities in the reference sentence by using additional context. We present results on MT evaluation but our approach may also be beneficial in other Natural Language Generation (NLG) tasks where discourse phenomena are present (e.g paraphrasing, data to text generation, chatbots, etc).

In conclusion, we argue that the MT community (and possibly the greater NLG community) should adopt metrics—such as those presented in this work—which take document-level context into account. This would better align automatic metrics with human evaluation, where document-level judgements have been shown to be more discriminative than sentence-level judgements. We also recommend that future research in metrics explore novel ways to incorporate context.

References

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1304–1313, New Orleans, Louisiana. Association for Computational Linguistics.

Sheila Castilho, Maja Popović, and Andy Way. 2020. On context span needed for machine translation evaluation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3735–3742, Marseille, France. European Language Resources Association.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy. European Association for Machine Translation.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2021. A Statistical Analysis of Summarization Evaluation Metrics Using Resampling Methods. *Transactions of the Association for Computational Linguistics*, 9:1132–1146.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Patrick Fernandes, Kayo Yin, Graham Neubig, and André F. T. Martins. 2021. Measuring and increasing context usage in context-aware machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6467–6478, Online. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021a. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021b. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Zhengxian Gong, Min Zhang, and Guodong Zhou. 2015. Document-level machine translation evaluation with gist consistency and text cohesion. In *Proceedings of the Second Workshop on Discourse in Machine Translation*, pages 33–40, Lisbon, Portugal. Association for Computational Linguistics.

Mandy Guo, Yinfei Yang, Keith Stevens, Daniel Cer, Heming Ge, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2019. Hierarchical document encoder for parallel corpus mining. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 64–72, Florence, Italy. Association for Computational Linguistics.

Christian Hardmeier and Marcello Federico. 2010. Modelling pronominal anaphora in statistical machine translation. In *Proceedings of the 7th International Workshop on Spoken Language Translation: Papers*, pages 283–289, Paris, France.

Yuchen Jiang, Tianyu Liu, Shuming Ma, Dongdong Zhang, Jian Yang, Haoyang Huang, Rico Sennrich, Ryan Cotterell, Mrinmaya Sachan, and Ming Zhou. 2022. BlonDe: An automatic evaluation metric for document-level machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1550–1565, Seattle, United States. Association for Computational Linguistics.

Prathyusha Jwalapuram, Shafiq Joty, Irina Temnikova, and Preslav Nakov. 2019. Evaluating pronominal anaphora in machine translation: An evaluation measure and a test suite. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2964–2975, Hong Kong, China. Association for Computational Linguistics.

Yunsu Kim, Duc Thanh Tran, and Hermann Ney. 2019. When and why is document-level context useful in neural machine translation? In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 24–34, Hong Kong, China. Association for Computational Linguistics.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796, Brussels, Belgium. Association for Computational Linguistics.

Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. 2018. OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

António Lopes, M. Amin Farajian, Rachel Bawden, Michael Zhang, and André F. T. Martins. 2020. Document-level neural MT: A systematic comparison. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 225–234, Lisboa, Portugal. European Association for Machine Translation.

Sameen Maruf, Fahimeh Saleh, and Gholamreza Haffari. 2021. A survey on document-level neural machine translation: Methods and evaluation. *ACM Computing Surveys*, 54(2).

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020. Results of the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Lesly Miculicich Werlen and Andrei Popescu-Belis. 2017. Validation of an automatic metric for the accuracy of pronoun translation (APT). In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 17–25, Copenhagen, Denmark. Association for Computational Linguistics.

Mathias Müller, Annette Rios, Elena Voita, and Rico Sennrich. 2018. A large-scale test set for the evaluation of context-aware pronoun translation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 61–72, Brussels, Belgium. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Ricardo Rei, Ana C Farinha, Chrysoula Zerva, Daan van Stigt, Craig Stewart, Pedro Ramos, Taisiya Glushkova, André F. T. Martins, and Alon Lavie. 2021. Are references really needed? unbabel-IST 2021 submission for the metrics shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1030–1040, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Yves Scherrer, Jörg Tiedemann, and Sharid Loáiciga. 2019. Analysing concatenation approaches to document-level NMT in two different domains. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 51–61, Hong Kong, China. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Damien Sileo, Tim Van De Cruys, Camille Pradel, and Philippe Muller. 2019. Mining discourse markers for unsupervised sentence representation learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3477–3486, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Y. Tang, C. Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *ArXiv*, abs/2008.00401.

Brian Thompson and Philipp Koehn. 2020. Exploiting sentence order in document alignment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5997–6007, Online. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020a. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121, Online. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020b. Paraphrase generation as zero-shot multilingual translation: Disentangling semantic similarity from lexical and syntactic diversity. In *Proceedings of the Fifth Conference on Machine Translation*, pages 561–570, Online. Association for Computational Linguistics.

Antonio Toral. 2020. Reassessing claims of human parity and super-human performance in machine translation at WMT 2019. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 185–194, Lisboa, Portugal. European Association for Machine Translation.

Billy T. M. Wong and Chunyu Kit. 2012. Extending machine translation evaluation metrics with lexical cohesion to document level. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1060–1068, Jeju Island, Korea. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *The 8th International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia. OpenReview.net.

Searching for a higher power in the human evaluation of MT

Johnny Tian-Zheng Wei[*]
University of Southern California
jtwei@usc.edu

Tom Kocmi and **Christian Federmann**
Microsoft
{tom.kocmi,chrife}microsoft@.com

Abstract

In MT evaluation, pairwise comparisons are conducted to identify the better system. In conducting the comparison, the experimenter must allocate a budget to collect Direct Assessment (DA) judgments. We provide a cost effective way to spend the budget, but show that typical budget sizes often do not allow for solid comparison. Taking the perspective that the basis of solid comparison is in achieving statistical significance, we study the power (rate of achieving significance) on a large collection of pairwise DA comparisons. Due to the nature of statistical estimation, power is low for differentiating less than 1-2 DA points, and to achieve a notable increase in power requires at least 2-3x more samples. Applying variance reduction alone will not yield these gains, so we must face the reality of undetectable differences and spending increases. In this context, we propose interim testing, an "early stopping" collection procedure that yields more power per judgment collected, which adaptively focuses the budget on pairs that are borderline significant. Interim testing can achieve up to a 27% efficiency gain when spending 3x the current budget, or 18% savings at the current evaluation power.

1 Introduction

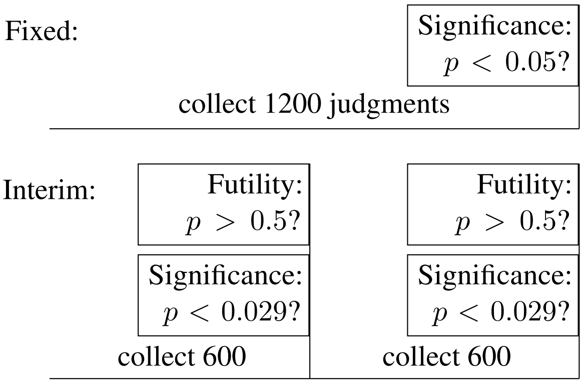


Figure 1: A graphical representation of evaluation with different testing procedures. Currently, our evaluation uses fixed testing, and our current budgets (depicted) often result in underpowered comparison (§5). To get a notable increase in power, we will need to spend more (§6), and interim testing is a way to spend efficiently. Interim testing allows for early stopping by trading off power for additional peeks. In MT, such a tradeoff is a favorable and can yield more power per judgment (§7).

In machine translation (MT), pairwise evaluations are conducted to identify the better system over a test domain. MT has long taken intrinsic quality as an object of interest, and assumes it can be determined directly from the output (Gatt and Krahmer, 2018). Most practitioners accept that human judgments reflect such quality, and take human evaluation as the gold standard (Bojar et al., 2016). In conducting an evaluation, the experimenter must allocate a budget to collect human judgments, and so evaluation can be an expensive endeavor. No one in the history of MT research has ever been satisfied with the cost or reliability of human evaluation (Graham et al., 2017; Chaganty et al., 2018; Saldías Fuentes et al., 2022, inter alia). Likewise, we were keen to find savings, upon the foundation of statistically rigorous inference.

Evaluation is a noisy process, and we may not expect a repeat experiment to declare the same winners. For one, we may want a holistic answer of the best system over the entire test domain, but we can only evaluate on a small and finite set of input source sentences (Koehn, 2004; Dror et al., 2018). This introduces a sample bias that our conclusion must be wary of. For another, human judgments on the same output may diverge, so we assume that humans are only a noisy reflection of the true intrinsic quality (Graham et al., 2015). This introduces additional noise when drawing a conclusion from our observations. Intuitively, using a larger test set or averaging over more human judgments should yield more consistency in pairwise comparison.

Inferential statistics is necessary in MT evaluation to declare "winning" MT systems under un-

[*] Work done at Microsoft.

certainty. Basic usage of statistical testing covers the use case of pairwise MT system comparison (Mathur et al., 2020). After data collection is complete, we can declare significance by computing a p-value (statistical primer in §3). When the p-value is low, a real effect is likely to exist. When the p-value is high, repeat experiments will be inconsistent (effectively tossing a coin), and no good decisions can be made even if you used the gold standard Direct Assessment (DA; Graham et al., 2015) annotation. Significance is the meta-analysis that guards against falsely declaring winners due to noise, with some level of guarantee.

Our work takes the perspective that the basis of solid comparison is in achieving significance. The rate/likelihood an experiment will observe significance is the *power*, and we would like it to be high. At the same time, we would like to minimize human effort and keep costs low. This paper investigates several aspects of the relationship between power and cost in human evaluation:

1. *How can we reason about the power of an evaluation?* We recommend a sensitivity perspective to evaluation, where we characterize an evaluation by its minimum detectable effect (MDE), or the smallest pairwise difference the evaluation will reliably yield significance on. By retrospectively analyzing significance in pairwise comparisons, we can derive an empirical MDE. **Our evaluations can reliably detect up to 2-3 point of DA difference, but comparisons often exhibit even smaller differences.**

2. *How can we notably increase the sensitivity of an evaluation?* With the appropriate power analysis, we can get a rough estimate of the number of samples required to achieve an acceptable sensitivity. To increase the sensitivity to the desired level, we might hope variance reduction techniques can give us the necessary sample efficiency. **If we wanted half of the past comparisons to reliably achieve significance, we needed at least 2x more samples, far beyond the ~1.2x sample efficiency variance reduction offers.**

3. *How can we spend more money efficiently?* If we are not satisfied with the power of our current evaluation, increasing the budget and collecting more judgments is necessary. Crucially, if we accept that small differences can't be known, our evaluation can be more efficient by focusing the budget elsewhere. **We verify that an "early stopping" procedure (interim testing) can can achieve up to a 27% efficiency gain when spending 3x our current budget, or 18% savings at our current evaluation power.**

2 Related work

There is a tradition of using test sets to estimate system performance over the general domain in machine learning (Hastie et al., 2001). There have been calls for statistically rigorous evaluation in natural language processing using significance testing (Dror et al., 2018), however its adoption in reporting has been mixed. For a classic task such as part-of-speech tagging, evaluation is generally significant/consistent even for small gains (Gorman and Bedrick, 2019). In MT, even moderate differences in metric gains (e.g. DA, MQM) may not be consistent, so there is a stronger need for significance testing. Historically, MT evaluation has been heavily based on statistical significance (Koehn, 2004).

MDEs have been used to describe the power of experiments in contexts such as education (which program results in increased test scores?) and sociology (Bloom, 1995). Berg-Kirkpatrick et al. (2012) empirically investigate the conventional wisdom that a certain metric gain corresponds to significance (e.g. 0.5 for BLEU). This threshold is exactly an evaluation's MDE. They find that a threshold has strong empirical backing, but a few experimental parameters affect this threshold. In our work, we propose taking a sensitivity perspective to evaluation, and reporting the expected MDE of an experiment instead of the other experimental parameters.

Any statistical technique that reduces the cost of human evaluation is, in another view, improving the power offered by some fixed budget. Chaganty et al. (2018) first proposed applying control variates to human evaluation. Control variates increase the sensitivity of an evaluation by leveraging information from a metric. This formulation conveniently allows us to analytically understand its performance based on the experimental conditions. In realistic experimental conditions, they found that the sample efficiency gain is at most 20%, which is in line with results reported in MT (Saldías Fuentes et al., 2022). Mendonça et al. (2021) propose using online learning to adaptively spend the evaluation budget on determining the best MT systems. However, their technique lacks in statistical rigor for decision making.

Knowing when to "early stop" an evaluation allows us to adaptively spend the budget on difficult pairs and save on easily distinguished pairs. It is known that peeking at the p-value while data collection is ongoing is problematic. Peeking inflates the chance of observing significance and the chance that such significant observation is incorrect (Albers, 2019). While always valid p-values can be calculated that adjust for this error and can be reported at any time, they are mathematically difficult to apply (Johari et al., 2015). Interim testing has been used in medical trials, where experimenters have an ethical consideration in stopping the experiment early (O'Brien and Fleming, 1979). By planning the number of peeks in advance, interim testing can offer rigorous statistical inference while potentially saving time and effort, packaged in an easy to understand technique (Lakens et al., 2021). Our work investigates whether the tradeoff between power and savings is favorable for MT evaluation.

3 A primer on inferential statistics

We consider pairwise comparisons as the basic unit of evaluation echoing calls from Mathur et al. (2020) and Kocmi et al. (2021). Pairwise comparisons are more interpretable than correlations, and more practical for production deployment scenarios. In a pairwise comparison we test the difference between two systems A and B. If you were just to collect a number of DA judgments for each system and declare a winner, a repeat experiment could yield different results due to experimental noise.

A statistical test guards against making an incorrect conclusion due to experimental noise. To do this, we assume a null hypothesis (that A is better than B) and examine how likely we could have made observed our data under this assumption. There are two outcomes of conducting a test:

(i) there is evidence of a significant difference which rejects the null hypothesis, or

(ii) the evidence is insufficient and we are unable to reject the null hypothesis.

In the case of (i), a significance test usually guarantees a false detection rate of at most α, where usually $\alpha = 0.05$. Therefore, the best outcome of statistical testing is the presence of significance, where our inferences enjoy a low false detection rate. The rate at which we can declare significance is called an experiment's *power* (typically denoted as $1-\beta$, where β is the false negative rate). In pairwise comparison, our evaluation should have an accuracy $(1-\alpha)(1-\beta)$ against the true, pairwise judgment.[1]

Intuitively, statistical testing can be loosely thought of as reducing the width of two confidence intervals, spaced by the true system difference of A and B (Krzywinski and Altman, 2013). The power of an experiment is then a function of these three aspects:

(a) First, the true system difference plays a role in the power. When the distance between the true scores is large relative to the noise, noise is unlikely to obfuscate the true pairwise ranking of the systems.

(b) Second, the variance of human judgment. The larger the variance in a single judgment, the more judgments that will be needed in an average to get a consistent estimate.

(c) Finally, the sample size or the budget. The number of judgments you collect shrinks the confidence intervals by a factor of $\sqrt{N}$ from the single judgment variance.

The more judgements you can collect the smaller these confidence intervals will be. When the confidence intervals don't overlap, the comparison is likely to achieve significance. These three factors all play a role in whether the intervals will be narrow enough.

If we know two of (a), (b), or (c), we can use the appropriate power analysis to deduce the third. Typically, we will observe the (b) human judgment variance, and make a guess at what the true difference (a) would be, to compute what (c) the budget we would have to spend is. When providing estimates for the budget, we would provide estimates under a range of guesses at what the true difference is (Card et al., 2020). Alternatively, we may also ask what the minimum detectable effect is for some fixed budget. Wei and Jia (2021) conducted power analysis in MT and found that small differences

[1] This pairwise accuracy holds if you assume that different MT systems always have different quality. By randomizing the systems, the null hypothesis will be true exactly half of the time.

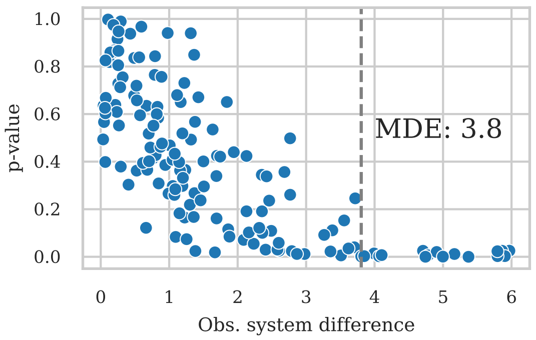


Figure 2: The minimum detectable effect (MDE) is illustrated in the ENU → FRA language pair. Each point represents a pairwise comparison conducted for this language pair. When evaluating pairs exhibiting differences larger than the MDE, 95% of pairs will achieve significance at the $\alpha = .05$ level, which totals to a pairwise accuracy of 90%. Unfortunately, most pairs are on the left hand side of this line. This is also the case for many other language pairs in the ShipData.

	Significant / insignificant	Obs. MDE	Median difference
ENU → FRA	30 / 153	3.8	1.2
ENU → DEU	19 / 151	3.5	0.7
FRA → ENU	3 / 140	2.4	0.6
DEU → ENU	27 / 130	1.9	0.6
JPN → ENU	78 / 127	2.9	3.2
ENU → JPN	40 / 94	3.8	1.8
ITA → ENU	2 / 81	2.8	0.5
CHS → ENU	30 / 78	2.6	1.5
ENU → PTB	28 / 74	1.0	0.6
ENU → SVE	31 / 73	4.4	1.4

Table 1: Significance and MDE results in the top-10 language pairs (by number of comparisons). Significance is calculated at the $\alpha = 0.05$ level. Observed MDEs are calculated for 90% pairwise accuracy. The median system difference is observed from the data. For most language pairs, less than half of the pairs had a significant observation. MDEs are small but most of the system differences appear to be even smaller.

require an infeasible amount of budget. This gives a hint that most of our MT evaluation is underpowered. Consistently conducting underpowered experiments run the risk of inflating the error rate in significant observations (Ioannidis, 2005).

4 Dataset

MT evaluation has an established tradition of conducting human evaluation and releasing public datasets. At the time of writing, the current annotation method of choice in MT is *Direct Assessment* (DA; Graham et al., 2015; Akhbardeh et al., 2021). Direct Assessment asks annotators to rate a translation's quality on a sliding point scale from 0-100. We study the **ShipData** presented in Kocmi et al. (2021), which is the largest human evaluation dataset of pairwise comparisons, accumulated over two years from internal evaluation campaigns at Microsoft Translator. No text is contained i.e. source, references, or outputs, but the raw DA scores are sufficient for our purposes. We focus on this dataset because it is large and often contain comparisons between state-of-the-art systems. It contains 4004 pairwise comparisons between two systems, where each system pair contains about 600 human judgments per system (1200 for both systems).

5 The sensitivity approach to evaluation

The basis of solid comparison is significance. Therefore, we need a way to reason about the power of an experiment. In this section, we recommend a sensitivity approach to evaluation, and retrospectively deduce the power of previous evaluations. By looking at the observed effect sizes we can also set a meaningful target power.

5.1 Minimum detectable effects (MDEs)

The pairwise evaluation of two MT systems is not a one-size fits all procedure, even though the MT literature uses a consistent annotation method (Federmann, 2018). Rather, an evaluation is our best attempt to answer which MT system is better with the evaluation annotation budget at hand. How much budget to allocate should depend on the circumstantial factors. Statistical inference can give us a probabilistic answer to this question with whatever evidence we are able to collect.

In the best case scenario, a significant result is observed and a winner is declared after the data is collected. However, significance depends on the conditions of the experiment (see §3), where the size of the pairwise difference, annotation variance, and number of samples all play a role. The pairwise difference and annotation variance are determined by the annotation method. Since most prefer to use a widely accepted annotation such as Direct Assessment (DA; Graham et al., 2015), these are factors we may not be able to change. However, we can increase the budget, and the larger the budget, the more likely we will be able to achieve significance for some fixed difference.

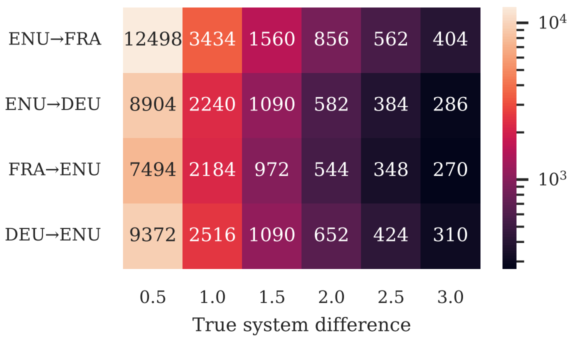


Figure 3: Power analysis for the total number of judgments required to achieve an MDE with 90% inference accuracy. These figures are calculated through simulation with distributional assumptions on the human scoring function (see §6.1). Compared to the observed MDEs, figures here serve as a lower bound. As the differences decrease linearly, the number of samples required increases exponentially.

		Variance (std. dev.)	Reducible variance
WMT21	`⋆-en`	866.2 (29.4)	23.1%
pSQM	`zh-en`	683.2 (26.1)	9.8%
pSQM	`en-de`	705.4 (26.5)	53.4%

Table 2: Total annotation variance and the reducible proportion of that variance. pSQM scores are provided by Freitag et al. (2021) and are collected from professional annotators. WMT21 scores are provided by Akhbardeh et al. (2021) and are collected from crowdworkers. pSQM scores are normalized from 0-100 for ease of interpretation. At least half of the variance is irreducible.

ρ	WMT21	pSQM(zh-en)	pSQM (en-de)
1.0	1.30	1.20	4.33
0.5	1.06	1.12	3.09
0.2	1.01	1.11	2.94

Table 3: Data efficiencies for the control variates estimator under different conditions. Each column represents a different condition of reducible variance, instantiated from observed statistics from Table 2. ρ is the correlation of the metric that would be used in the control variates estimator. With the exception in pSQM `en-de`, variance reduction is far from giving us the 2x-10x multiplier we need.

We recommend to think about an MT evaluation in terms of its sensitivity. With a fixed budget and annotation method, there is some deducible minimum detectable effect (MDE; Bloom, 1995), where evaluating differences larger than the MDE will enjoy a comfortable level of power (rate of significance). Alternatively, if we did not observe significance for some experiment, we may suspect that the true difference is likely to be lower than the experiment's MDE. With a sensitivity perspective, our consideration is now to conduct DA evaluations with a budget large enough to exhibit an appropriate MDE. Ideally, our evaluation exhibits an MDE small enough where we believe any smaller differences are not practically meaningful (more in §6.1). Realistically, we would set up an evaluation with MDEs as small as our budgets allow.

5.2 Observed MDEs

In this section, we attempt to retrospectively understand the MDEs/sensitivity of our past evaluations. Refer to Figure 2 for graphical intuition. We can empirically estimate (as opposed to making assumptions and simulating, see Card et al., 2020) an (observed) minimum detectable effect by sorting all the pairs by their observed absolute system difference, and choosing the difference where comparisons with a larger system difference (effect size) will have at least 95% of experiments showing significance (corresponding to experimental power $1 - \beta = 0.95$) at a level of $\alpha = 0.05$ by the Mann Whitney U (MWU) test. This ensures that at least $(1 - \alpha)(1 - \beta) \approx 0.9$ of the pairs should be accurate (Wei and Jia, 2021). We can interpret this as the threshold at which our experiments will stop being accurate at the 90% level.

The minimum detectable effects (MDE) are small, but differences between systems are even smaller. Refer to Table 1. Our evaluations have been able to detect up 1 or 2 points of system-level DA difference, but often a third of the comparisons are still not significant. Looking at the density of the differences (see the x-axis in Figure 2) we see that most of the pairs exhibit small differences. An immediate consequence is that most of the budget is being spent to declare ties. Most of our comparisons are underpowered, and where the p-value is high the experiments are not much better than a coin toss. The median difference provides a target MDE if we want half of our evaluations to show significance (alternatively, declaring ties in half of the evaluations is acceptable).

6 Known unknowns

Now that we have established a way to reason about experimental power, we conduct power analysis to understand how much more gain we need to improve our power to a desired sensitivity. We investigate whether variance reduction techniques are sufficient, and conclude that the only way forward is to increase the annotation budget.

6.1 Power analysis for the desired sensitivity

As suggested in Card et al. (2020), we can roughly determine the number of samples for a fixed power using simulation. As with any power analysis, we must make some assumptions to estimate the number of samples needed. Here we assume that the judgments for a given system's translation is distributed as $s \sim 100 - \text{Gamma}(k, \theta)$ where $k = \frac{\mu^2}{\sigma^2}$ and $\theta = \frac{\sigma^2}{\mu}$ are fit to match the average mean and variance of a system for that language pair. We choose the use of the Gamma distribution because the resulting scoring distribution is such that most of the scores are high, and the more severe the translation error the more rare it is, which matches what we observe in Kocmi et al. (2021). We then use the bisection method to determine the integer whose power has the closest match to our desired β value. We find that the simulation reasonably matches empirically observed MDEs.

Power analysis shows that most pairs needs not a little, but a lot more judgments. Refer to Figure 3. Comparing to the observed MDEs, the power analysis is *optimistic*, where the figures we provide can be seen as a lower bound. Even a reduction of our MDE to 1 point can require up to 2x times more judgments (than originally used in the ShipData). We highlight the fact that as differences get linearly smaller, the number of samples is an exponential growth. The nature of statistical estimation is that smaller differences are increasingly elusive.

In the search for higher power, we must also keep in mind that arbitrarily small differences require arbitrarily large budgets. Therefore, for modern state-of-the-art comparisons, some differences will be left unknown. We can not fantasize about detecting every single small difference out there just by spending more budget or applying some strong statistical technique (see §6.2). Perhaps this may be taken in stride, as mathematicians learned to accept the existence of unprovable theorems nearly a century ago (Gödel, 1934). Many other important fields such as domain adaption also grapple with their unknowns (Ben-David et al., 2010).

6.2 Variance reduction is inadequate

Generally, we assume that a human evaluator scores a segment with the true segment level quality score, plus some noise. If $H(x)$ is the human scoring function on system translations x, there are 2 parts to the scoring variance. We can decompose the variance of H to

$$\begin{aligned} \text{Var}(H(x)) = \mathbb{E}[\text{Var}(H(x)|x)] \\ + \text{Var}(\mathbb{E}[H(x)|x]) \end{aligned} \quad (1)$$

by the law of total variance. The first part is the variance of the true translation quality scores, capturing the real difference in quality across output translations, and the second part is the rest of the variance. The second term, which we broadly term annotator noise, can include annotator biases, preferences, and even mood.

Using repeat judgments we can estimate the second term (annotator noise), which is similar to an inter-annotator agreement (Wei and Jia, 2021). Since the ShipData doesn't contain any repeat judgments, we provide estimate of the second term from a few similar datasets (Akhbardeh et al., 2021; Freitag et al., 2021). Refer to Table 2. In designing variance reduction techniques, we usually leverage metric scores to reduce the first term, but not annotator information to reduce the annotator noise (second term), as it is too difficult (Saldías Fuentes et al., 2022).

With variance reduction (VR) techniques, we can achieve a higher power with the current budget by leveraging side information (Owen, 2013). However, VR is not arbitrarily powerful, and its effectiveness is constrained by the amount of reducible variance present, and how much of the reducible variance you can actually reduce. Here, we look at the control variates technique[2] which leverages the linear information in a metric for the estimation of system quality. The data efficiency in Chaganty et al. (2018) describes how many times a control variates estimator improves over the regular sample mean estimate, and is characterized by

$$\text{DE} := \frac{\text{Var}(\hat{\mu}_{\text{mean}})}{\text{Var}(\hat{\mu}_{\text{cv}})} = \frac{1+\gamma}{1-\rho^2+\gamma} \quad (2)$$

[2] Equal proportion stratified sampling is a special case of control variates, so these results also apply (Owen, 2013). Any technique which uses a metric to bin outputs, where the same number of outputs are sampled for scoring within each bin, are constrained by these results as well.

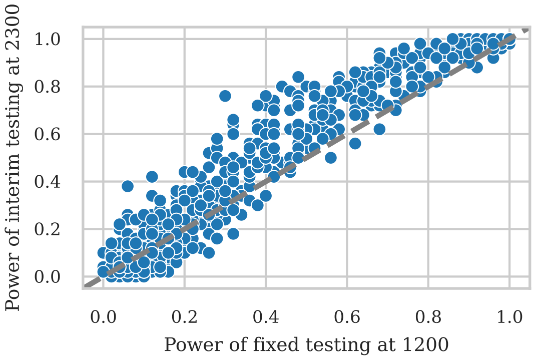


Figure 4: The average power of each pairwise comparison for fixed testing at 1200 against interim-futility testing at 2300. Each point represents a pairwise comparison. When planning for 2300 judgments with interim-futility, the actual amount of judgments collected in our simulation is about 1200. For the same budget, we see that interim-futility testing boosts the power of moderate to high-powered pairs, but drops that of the lower powered pairs.

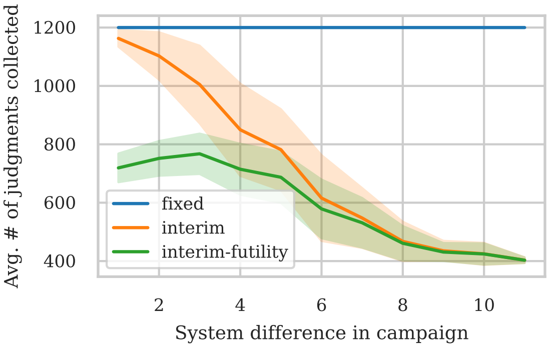


Figure 5: The average number of judgments collected by each sampling method. For interim and interim-futility, 1200 judgments were planned, and the actual judgments collected are strictly less. As the system differences grow larger, both methods have the potential to stop early. For interim-futility, pairs with small differences also incurred less judgments.

where ρ is the sentence-level Pearson correlation of the metric and

$$\gamma = \frac{\sigma_a^2}{\sigma_f^2} = \frac{\mathbb{E}[\text{Var}(H(x)|x)]}{\text{Var}(\mathbb{E}[H(x)|x])} \tag{3}$$

Refer to Table 3. **With the optimistic assumption of a perfect metric, we often only get a ~1.2x efficiency gain from VR, far from the 2-10x multiplier we need to obtain significant comparisons.** The gains we predict for VR is consistent with the practical results presented in Saldias et al. (2022). These reduction techniques work, but is far from achieving what we need, echoing the narrative of Chaganty et al. (2018).

7 Spending effectively

To have a notable gain in sensitivity, variance reduction alone is inadequate. Therefore, spending is necessary in the search for higher power. This section describes a simple yet statistically rigorous way of "early stopping" in a human evaluation campaign. Interim testing adaptively allocates the budget to borderline significant pairs, and can be seen as an efficient way to spend.

7.1 Peek-a-boo! Planning interim peeks

Savings can be achieved if we can stop data collection as soon as a result can be concluded. If the experimenter runs the preferred statistical test (at false detection rate $\alpha = 0.05$) periodically while data collection is on-going, the final process will have a false detection far higher than the α intended (Albers, 2019). There are a class of sequential sampling techniques, which allow you to test after every single sample while maintaining the false detection rate constant, but are mathematically difficult to apply (Johari et al., 2015).

A simpler solution is to use interim sampling and apply a correction for multiple testing (Lakens et al., 2021). For instance, the Pocock correction (Pocock et al., 1987) is appropriate when multiple comparisons are made, but we want a false detection to be maintained at a desired α.[3] Refer to Figure 1. For interim testing, we can plan in advance to collect batches of data, and test between each batch. To maintain a final false detection rate to the fixed procedure, your interim tests must have an α_0 appropriately adjusted with the Pocock correction. The downside is that this correction is conservative, and each test has less power.

At each interim point, we can also stop for futility, or when we see that even in completion of the data collection, we are unable to achieve significance. Practically, there are many ways to set up this stopping rule (Lakens et al., 2021), but in our simulation we find that a simple heuristic (checking if the $p > 0.5$) works well for our purposes. An alternative view of futility stopping is that we are unwilling to conduct the analysis of the original

[3] Here's why we need a correction: imagine 20 comparisons made at $\alpha = 0.05$ where the null hypothesis is true, then the probability of getting at least 1 significant result is actually $1 - 0.95^{20} \approx 0.63$.

experiment with the corresponding MDE.

7.2 Experimental setup

We compare three different kinds of testing methods. Refer to Figure 1.

- **Fixed** testing is most commonly used in evaluation. In fixed testing, the annotation budget is spent all at once, and the statistical test is performed at the end. The advantage of fixed testing is that only statistical test is performed with the highest (least conservative) alpha threshold (e.g. is $p < 0.05$?).
- **Interim** testing plans to spend the budget in equal sized steps, with an interim analysis between each step. If significance is observed at any point, the data collection is terminated. We always plan for 3 peeks, and use the Pocock correction (e.g. is $p < 0.0221$? at each peek). While the testing threshold is lower (more conservative), the savings obtained from some pairs can be used on others, by planning more judgments for all pairs.
- **Interim-futility** is the same as interim testing but also applies a futility stopping rule at each analysis. If $p > 0.5$ then the experiment is terminated early. Futility stopping does not affect the false detection rate so it does not need to be adjusted. Futility stopping results in strictly less power, but the savings can be used elsewhere, by planning more judgments.

To benchmark these testing procedures against each other, we simulate data collection from the pairs in the ShipData by sampling with replacement. For each pair we simulate each testing procedure 1000 times and record the number of times the procedure is able to achieve significance. For all tests we use the Mann Whitney U test (standard to machine translation; Akhbardeh et al., 2021) with a testing threshold of $\alpha = 0.05$. Within the ShipData, each pair only has about 1200 judgments, from which we often oversample. We note that this is our best faith attempt to study these testing methods in the large budget regime, and actual benchmarking would require infeasible cost, so the simulation can serve as our best synthetic testbed.

7.3 Results

Refer to Figure 5. For a fixed sampling procedure, the number of samples collected is constant for every effect size. This can be inefficient as pairs with large differences do not need as many judgments to declare significance. Interim testing is adaptive; as the differences get larger, interim testing can declare significance at an early step. For interim-futility, less judgments are also collected for the pairs with the smallest differences, where early steps may declare futility. We will later see that the interim-futility behavior is most favorable.

Refer to Figure 4. When comparing fixed and interim-futility, we compare two procedures that spend the same budget. Since interim sampling spends more on borderline pairs, the power for pairs with moderate to high differences increases. Savings are made on pairs with both large and small differences, with small difference pairs having a decrease in power. We highlight that interim-futility is a different kind of testing. **While the use of fixed testing seeks to best detect every difference no matter how small, the use of interim-futility prioritizes the pairs that have borderline significant differences.**

Refer to Figure 6(a). The main metric we benchmark these methods is by the average power, or the number of significant comparisons over all the ShipData. When comparing over all pairs, interim testing has slightly better performance, but interim-futility gives considerable gains even at current budget sizes. Our results show that to attain the fixed testing power at 1200, interim testing only needed to spend 990 judgments per comparison, which is an 18% saving[4]. The advantage of interim sampling over fixed sampling is even more pronounced when we are spending large budget sizes, where we can gain 28% savings at 3600 judgments (3x). Refer to Figure 6(b). When testing small differences interim sampling is worse than fixed sampling, as it has a stricter significance threshold. However, interim-futility is able to stop on pairs with little hope and prioritize the borderline significant pairs. Refer to Figure 6(c). On pairs with large differences interim sampling is best, with interim-futility achieving similar performance. For large differences futility stopping should rarely trigger, so the two methods should be similar.

We want to highlight that the distribution of the differences is key to the success of the interim-futility testing procedure. Since most of the pairs are concentrated either in the dense

[4] All the results in this paragraph are derived using linear interpolation, akin to using a ruler on Figure 6.

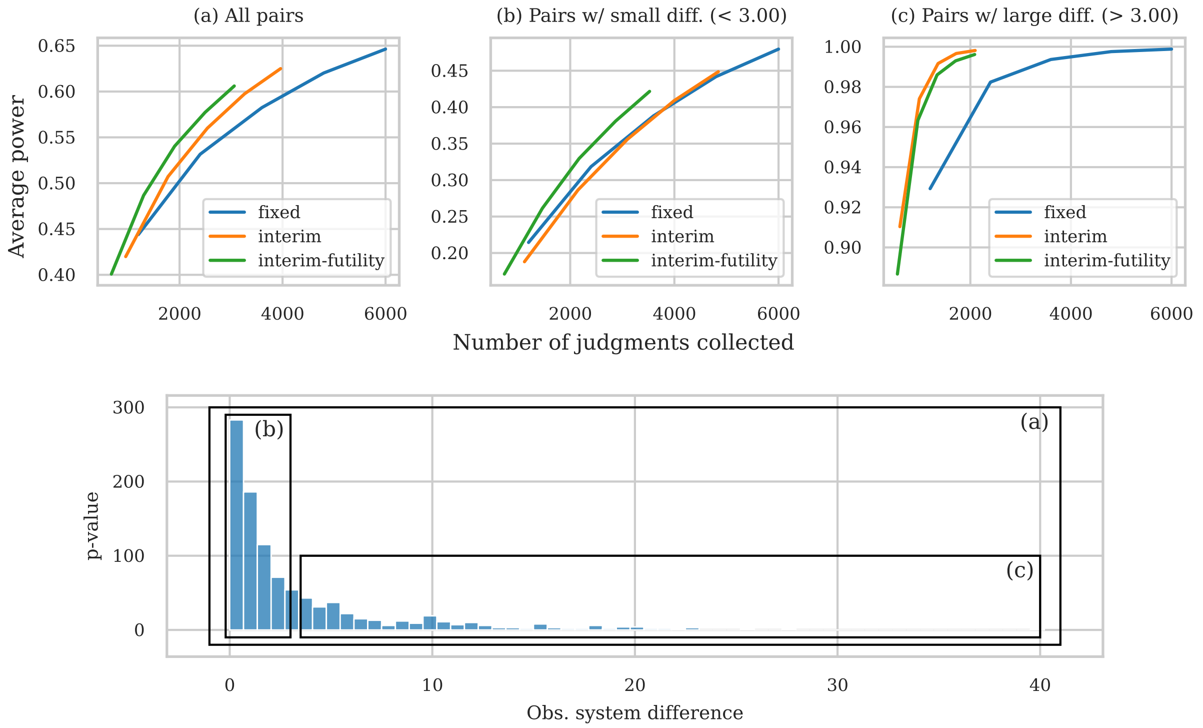


Figure 6: (Top) The average power of each testing procedure across the ShipData for different sized budgets. (a) Shows the average power across all data, and (b) shows it over pairs with large differences and (c) shows it for small differences. (Bottom) The histogram of the true differences in each pairwise comparison. These are true differences due to the simulation we used to test these procedures. Interim-futility is most favorable by average power in (a), (b) and (c). Interim testing is weaker in (c) due to its stricter significance threshold.

region of small differences or in the long tail of large differences, these are areas where interim-futility can early stop. Compared to fixed testing, interim-futility will be able to make savings here to spend elsewhere. Crucially, the application of futility stopping also requires a change in our evaluation mindset, as we must be willing to accept that some small differences are not worth detecting. If we can make this change, then interim-futility is most favorable in terms of average power.

8 Limitations

The most important assumption of our work is in the use of Direct Assessment (DA). While our methods can generalize to any real valued judgment, we analyzed DA because of its widely recognized, gold standard status in MT evaluation. DA is a particularly noisy judgment, and so the power and variance reduction results are pessimistic. However, we believe that the study of annotation will be the most important direction in MT evaluation.

Let's take Hassan et al. (2018), where one of the first claims of MT-human parity was made. By their evaluation, which was conducted according to the community standard, no significant difference was found between human and machine translations with a reasonable budget, and so a tie was declared. Toral et al. (2018) reassesses this claim, and essentially presents a series of alternative evaluations and observe significant differences that contradict with Hassan et al. (2018). This is just one of many studies which compels an alternative evaluation with qualitative insight (Läubli et al., 2018, 2020; Freitag et al., 2021).

Our perspective is that significance is only one pillar of MT evaluation. It is our hope that the analyses in this work will further our understanding of significance and evaluation power. However, the second pillar of MT evaluation is in the annotation method. While power is quantitative, the study of annotation methods will be qualitative. Going forward, understanding how we can change the annotation method to increase the power will be crucial. We will need good qualitative understanding to be able to move away from DA and establish new gold standards.

In addition, we showed that interim testing is only effective for pairwise comparison. Future work should look to make savings in the leaderboard styled evaluation of WMT. This may come in the form of generalizing interim sampling for multiple comparisons or formalizing the bandit results from Mendonça et al. (2021) in terms of statistical inference.

9 Conclusion

Our work is motivated by the cost of human evaluation in machine translation. Before searching for a higher power from our current budget, we determined how much more power was necessary. In doing so, we recommend taking a sensitivity approach to evaluation. From here we came to the conclusion that to achieve the power/sensitivity necessary, variance reduction alone would be insufficient, and spending is our only option. If we decide to allocate larger budgets, interim testing is a more effective way to spend, which can yield 18% savings at the current evaluation power, or 27% savings at 3x the original budget.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (wmt21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Casper Albers. 2019. The problem with unadjusted multiple and sequential statistical testing. *Nature Communications*, 10(1):1–4.

Shai Ben-David, Tyler Lu, Teresa Luu, and Dávid Pál. 2010. Impossibility theorems for domain adaptation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, volume 9 of *JMLR Proceedings*, pages 129–136. JMLR.org.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005, Jeju Island, Korea. Association for Computational Linguistics.

Howard S. Bloom. 1995. Minimum detectable effects: A simple way to report the statistical power of experimental designs. *Evaluation Review*, 19(5):547–556.

Ondřej Bojar, Christian Federmann, Barry Haddow, Philipp Koehn, Matt Post, and Lucia Specia. 2016. Ten years of wmt evaluation campaigns: Lessons learnt. In *Proceedings of the LREC 2016 Workshop "Translation Evaluation–From Fragmented Tools and Data Sets to an Integrated Ecosystem*, pages 27–34.

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. 2020. With little power comes great responsibility. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9263–9274, Online. Association for Computational Linguistics.

Arun Tejasvi Chaganty, Stephen Mussmann, and Percy Liang. 2018. The price of debiasing automatic metrics in natural language evalaution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 643–653. Association for Computational Linguistics.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Christian Federmann. 2018. Appraise evaluation framework for machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 86–88, Santa Fe, New Mexico. Association for Computational Linguistics.

Markus Freitag, George F. Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *CoRR*, abs/2104.14478.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Intell. Res.*, 61:65–170.

Kyle Gorman and Steven Bedrick. 2019. We need to talk about standard splits. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2786–2791, Florence, Italy. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, and Nitika Mathur. 2015. Accurate evaluation of segment-level machine translation metrics. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1183–1191. The Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2017. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, 23(1):3–30.

Kurt Gödel. 1934. On undecidable propositions of formal mathematical systems.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation. *CoRR*, abs/1803.05567.

Trevor Hastie, Jerome H. Friedman, and Robert Tibshirani. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer.

John P. A. Ioannidis. 2005. Why most published research findings are false. *PLOS Medicine*, 2(8):null.

Ramesh Johari, Leo Pekelis, and David J. Walsh. 2015. Always valid inference: Bringing sequential analysis to a/b testing.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Martin Krzywinski and Naomi Altman. 2013. Error bars. *Nature Methods*, 10(10):921–922.

Daniel Lakens, Friedrich Pahlke, and Gernot Wassmer. 2021. Group sequential designs: A tutorial.

Samuel Läubli, Sheila Castilho, Graham Neubig, Rico Sennrich, Qinlan Shen, and Antonio Toral. 2020. A set of recommendations for assessing human-machine parity in language translation. *J. Artif. Intell. Res.*, 67:653–672.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has machine translation achieved human parity? A case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4791–4796. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Vânia Mendonça, Ricardo Rei, Luisa Coheur, Alberto Sardinha, and Ana Lúcia Santos. 2021. Online Learning meets Machine Translation evaluation: Finding the best systems with the least human effort. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3105–3117, Online. Association for Computational Linguistics.

Peter C. O'Brien and Thomas R. Fleming. 1979. A multiple testing procedure for clinical trials. *Biometrics*, 35(3):549–556.

Art B. Owen. 2013. *Monte Carlo theory, methods and examples*.

Stuart J Pocock, Nancy L Geller, and Anastasios A Tsiatis. 1987. The analysis of multiple endpoints in clinical trials. *Biometrics*, pages 487–498.

Belén Saldias, George F. Foster, Markus Freitag, and Qijun Tan. 2022. Toward more effective human evaluation for machine translation. *CoRR*, abs/2204.05307.

Belén Saldías Fuentes, George Foster, Markus Freitag, and Qijun Tan. 2022. Toward more effective human evaluation for machine translation. In *Proceedings of the 2nd Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 76–89, Dublin, Ireland. Association for Computational Linguistics.

Antonio Toral, Sheila Castilho, Ke Hu, and Andy Way. 2018. Attaining the unattainable? reassessing claims of human parity in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 113–123, Brussels, Belgium. Association for Computational Linguistics.

Johnny Wei and Robin Jia. 2021. The statistical advantage of automatic NLG metrics at the system level. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6840–6854, Online. Association for Computational Linguistics.

Test Set Sampling Affects System Rankings: Expanded Human Evaluation of WMT20 English–Inuktitut Systems

Rebecca Knowles and **Chi-kiu Lo**
National Research Council Canada (NRC-CNRC)
{rebecca.knowles,chikiu.lo}@nrc-cnrc.gc.ca

Abstract

We present a collection of expanded human annotations of the WMT20 English–Inuktitut machine translation shared task, covering the Nunavut Hansard portion of the dataset. Additionally, we recompute News rankings to take into account the completed set of human annotations and certain irregularities in the annotation task construction. We show the effect of these changes on the downstream task of the evaluation of automatic metrics. Finally, we demonstrate that character-level metrics correlate well with human judgments for the task of automatically evaluating translation into this polysynthetic language.

1 Introduction

Translation between Inuktitut[1] and English was featured as part of the 2020 News Translation task at WMT (Barrault et al., 2020). The English–Inuktitut machine translation system rankings published in Barrault et al. (2020) were incomplete, due to the delay in the 2020 annotation campaign and because they only cover the out-of-domain portion of the test set. In this work, we present:

- an expanded dataset of human annotations that covers the in-domain portion of the test set,[2]
- an analysis of both the existing (out-of-domain) human annotations and the new (in-domain) annotations,
- revised system rankings based on the new annotations and controlling for irregularities in the original data collection process,
- and correlations of automatic MT evaluation metrics with the revised system rankings and the newly collected human annotations.

Dataset	Segments
Hansard-A (H-A)	11404
Hansard-B (H-B)	7801
All Hansard	19205
WMT20-DA (N-1)	8000
WMT20-DA2 (N-2)	12002
WMT20-DACrowd (N-C)	9728
All News	29730
Total	48935

Table 1: Number of segments annotated in each dataset, without any data filtering. We use names corresponding to the dataset files, with short forms in parentheses.

Our aim with this work is to release a broader set of human annotations, for continued research on translation between English and Inuktitut, as well as to demonstrate the downstream impacts of irregularities in WMT data collection and publication. In particular, we draw attention to how errors in human annotation setup (not errors in the work of the annotators, but errors in the way that organizers constructed the annotation tasks) and the failure to account for their effects impact both the validity of the rankings themselves and the shared task on automatic metrics which relies on the rankings. In this way, the 2020 WMT task on Inuktitut serves as a case study of a broader issue in the field. We provide suggestions for how to account for irregularities in existing annotation task construction and release the code and data to replicate our results.

2 Data

The test set for the WMT 2020 English–Inuktitut shared task consisted of data from the Nunavut Hansard as well as from Nunatsiaq News, collected and shared with permission. The News test data consisted of documents collected from Nunatsiaq News between September and November, 2020, as was standard for the shared task. The Hansard data in the test set, however, was collected from earlier dates. The main training data available for building constrained systems was the Nunavut Hansard 3.0

[1] We use the term *Inuktitut* here because the website of the Legislative Assembly of Nunavut lists Inuktitut, Inuinnaqtun, and English (along with French) as the languages spoken in the House (https://assembly.nu.ca/faq#n113) and the version of the Hansard released as training data is the Inuktitut version, written in syllabics. See Appendix A.

[2] Dataset and code: https://github.com/nrc-cnrc/Reranking-WMT20-IKU

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 140 - 153
December 7-8, 2022 ©2022 Association for Computational Linguistics

	H-A	**H-B**	**N-1**	**N-2**	**N-C**	***Total***
A	-	-	2800	-	-	*2800*
B	-	-	-	3200	-	*3200*
C	-	3801	-	200	-	*4001*
D	5600	4000	4600	2000	-	*16200*
E	-	-	600	6602	-	*7202*
F	2403	-	-	-	-	*2403*
G	3401	-	-	-	-	*3401*
Un.	-	-	-	-	9728	*9728*

Table 2: Number of segments annotated by annotator (anonymous annotator ID shown in the first column, with Un. representing all unknown annotators who annotated the crowd data) and dataset.

(Joanis et al., 2020), consisting of aligned proceedings of the Legislative Assembly of Nunavut. As a consequence, the Nunavut Hansard test data could be considered "in-domain", while the News data was "out-of-domain" (the development data was similarly divided between the two domains).

The annotators who did the work of human evaluation discussed here were fluent language experts at the Pirurvik Centre,[3] paid at professional rates. All the human annotations were collected in the segment rating with document context (SR+DC) style of direct assessment (DA; Graham et al. 2013, 2014, 2016) using the Appraise interface (see Barrault et al. 2020 for additional interface details). For each segment, annotators viewed the source sentence and a candidate translation, and scored the translation on a pseudo-continuous sliding scale from 0-100. These segments were displayed in document context, and annotators also provided document scores (we omit those from this work).

Each News story had a unique document ID, but the Hansard data was treated as a single document containing 1566 lines. This had two main consequences with respect to human annotation of system outputs. The first and most obvious is that the annotations collected at WMT (on which the Findings paper's rankings (Barrault et al., 2020) were at least partially based) only included annotations of the News data (out-of-domain). The reason for this is that the code used to generate sessions of annotations in the SR+DC DA human annotation task structures pooled all documents and then sampled documents "at random (without replacement) and assigned [them] to the current HIT [human intelligence task] until the current HIT comprise[d] no more than 70 segments in total"; since the Hansard data was treated as one document with more than 70 segments, it was never sampled.[4] The second is that the News documents were longer on average than News documents for other language pairs. English–Inuktitut News documents ranged from 12 to 137 lines in length, with a mean of 39.0 (standard deviation 20.5) and a median of 36. Documents for other language pairs that were evaluated in the SR+DC format ranged from 2 to 32 lines in length, with a mean of 12.0 lines (standard deviation 6.2) and a median of 11. As a consequence, each annotation session of English–Inuktitut News data is less likely to contain documents translated by the full set of submitted systems (12 submitted systems and human reference), which has the potential to cause problems when scores are normalized per-annotation session (Knowles, 2021).[5] Additionally, a portion of the source and reference data News segments contained spurious quotation marks (see Appendix B). We now discuss the News and Hansard annotation processes.

2.1 News Annotations

There are several other noteworthy issues about the News data collection. As shown in Table 1, there are three direct assessment datasets collected at WMT that contain News-only annotations of English–Inuktitut translations. The first, N-1, consists of 8000 segments and does not contain any annotations of reference segments. The second, N-2, contains 12002 segments and *does* contain annotations of reference segments. Both of these were annotated by fluent language experts at the Pirurvik Centre. There is a third set of data, N-C, which was annotated by other annotators (the Findings paper does not clarify who those annotators were, so we will focus our analysis on the first two datasets, known to be collected through Pirurvik). The fact that one set of data was collected with reference segments included and the other was not

[3] https://www.pirurvik.ca/

[4] Note that we will use HIT and annotation session interchangeably in this paper. An annotation session that received a single ID and thus was used as the basic chunk of data for computing z-scores typically (but not always) consisted of two HITs, each containing 100 segments. However, the way the data is released, the two HITs are not distinguishable from one another, hence our reference instead to the annotation session. Note that an individual annotator may have completed many such sessions.

[5] The use of z-score carries with it an implicit assumption that the annotation session, HIT, or set of data annotated by one annotator is representative of the whole. The raw scores for human references tended to be near-perfect, while one system's scores were zero, meaning that whether an individual annotation session contained one, both, or neither of these would unduly influence the z-score computation.

also has the potential to cause problems in generating system rankings. Because system rankings from SR+DC run through Appraise are typically calculated based on z-scores computed at the annotation session level, and because these sessions are *not* representative of the distribution of systems, they will be erroneously standardizing out real differences in quality. Even if they did compute z-scores over annotators, we can see in Table 2 that not all annotators completed annotation sessions in each dataset, meaning that some annotated reference segments and some did not; again, this means that it is inappropriate to calculate z-scores in the standard WMT fashion (even over annotators instead of over sessions). The data collection also contained quality assurance segments (these are called "BAD" segments, and quality assurance is described in more detail in Appendix C).

The system submitted under the name zlabs-nlp (no corresponding paper submitted) consisted of the exact source (English) data, but was nevertheless included in the annotation tasks. The annotators from Pirurvik received instructions to give a score of 0 to output that was not in the target language (i.e., Inuktitut) and this is reflected in their scores (almost all 0 for zlabs-nlp segments),[6] while the scores for the Crowd annotation set are much more wide-ranging (indicating that those annotators may not have received the same instructions).

2.2 Hansard Annotations

Following the completion of WMT 2020, we collected annotations of the Nunavut Hansard portion of the test set. Like the News annotations, fluent Inuktitut-language experts from Pirurvik Centre performed these segment rating with document context (SR+DC) annotations using the Appraise interface; with the help of the shared task organizers, we collected data using the same web interface as was used for the News data, allowing us to keep that portion of the annotation process consistent.

The data was processed and collected with the following noteworthy changes.[7] First, data from zlabs-nlp (exact copies of the source text) were omitted from annotation, as those scores are not representative of translation. Second, the Hansard was manually divided into pseudo-documents, ranging in length from 8 lines to 26 lines, with an average of 14.6 (standard deviation 3.7) and median 15. This is closer to the average document length for other language pairs, and enables annotation sessions to contain a more diverse set of system/document pairs. Third, in this set of annotations, references – and all systems – were more evenly distributed across annotators, improving validity of the z-score assumptions (Knowles, 2021). Finally, as shown in Table 1, the Hansard annotations were split into two parts. Wishing to ensure that all systems were annotated on consistent sets of documents, but unsure as to whether annotator time and budget would cover the full Hansard test set, we first randomly split the set of documents in two, and then generated annotation sessions by sampling from one half (Hansard-A) or the other (Hansard-B). Fortunately, annotators completed all sessions. We did not include quality assurance segments in this task, as all annotators were known to be qualified (see Appendix C).

2.3 Systems

Twelve systems were submitted to the English–Inuktitut task. In alphabetical order by team name, they were: CUNI-Transfer (Kocmi, 2020), Facebook_AI (Chen et al., 2020), Groningen (Roest et al., 2020), Helsinki (Scherrer et al., 2020), NICT_Kyoto (no corresponding paper),[8] NRC (Knowles et al., 2020), OPPO (Shi et al., 2020), SRPOL (Krubiński et al., 2020), MultiLingual_Engine_Ubiqus (Hernandez and Nguyen, 2020), UEDIN (Bawden et al., 2020), UQAM_TanLe (no corresponding paper), and zlabs-nlp (no corresponding paper). Of these twelve, Barrault et al. (2020) listed MultiLingual_Engine_Ubiqus and UQAM_TanLe as unconstrained entries (meaning that they chose to use additional data outside of those provided for the constrained version of the shared task). All systems for which we have a description used Transformer models (Vaswani et al., 2017).

3 Approaches to Rankings

In this work, we will generate two sets of rankings: system rankings over the Hansard data and system rankings over the News data. While there would be reason to desire a single ranking that covers both in-domain (Hansard) and out-of-domain

[6] The 7 scores of 1 may be simply due to slider operation.

[7] This data collection was completed prior to the publication of Knowles (2021), and as such only addresses a portion of the concerns raised in that paper. We seek to address other concerns from that paper in our analysis of the data.

[8] The Findings paper cites Marie et al. (2020) for NICT_Kyoto, but that paper does not describe an English–Inuktitut MT system.

(News), that would raise the question of how to balance the two, and would also be a challenge to produce given the differences in the data collection processes. Having two rankings also highlights differences in performance across those domains.

The Hansard rankings come from the annotations that will be released alongside this paper, while the News rankings are a reranking based on the data collected at the WMT shared task. Here we discuss how the rankings computed for this paper differ from those produced at WMT. A partial description of the WMT rankings can be found in Barrault et al. (2020). The main issues we try to address in our new rankings are those raised in Knowles (2021) around the instability of rankings, particularly when the annotation sessions contain distributional issues that make the usual z-score computation inappropriate. We attempt to handle these issues both in proactive ways (through modifications to the "document" lengths and system distributions in the setup of the annotation of Hansard data) and in reparative ways (when we make use of the existing WMT News annotations).

3.1 Hansard Ranking Approach

Ave.	Ave.z	System
89.9	0.249	SRPOL
87.5	0.201	Groningen
88.6	0.192	NICT_Kyoto
88.8	0.170	NRC
88.1	0.160	Human-A
87.1	0.133	CUNI-Transfer
85.9	0.120	Facebook_AI
85.6	0.046	UEDIN
83.6	-0.055	Helsinki
78.0	-0.127	MultiLingual_Engine_Ubiqus
76.5	-0.360	UQAM_TanLe
65.6	-0.789	OPPO

Table 3: Hansard ranking, computed using the standard WMT approach. Unconstrained systems in grey. Horizontal lines separate significance clusters.

For the Hansard rankings (Table 3), we compute them as follows. We have a mapping between annotators and annotation sessions, so for each annotator, we collect all of the data from all of their annotation sessions. Given one annotator's full set of annotations, we compute the mean m_a and standard deviation s_a (where a is the annotator). These are then used to compute the z-scores for every segment that they annotated. Given a raw score x produced by annotator a, its z-score is:

$$z = \frac{x - m_a}{s_a} \tag{1}$$

After z-scores have been computed for all segments annotated by all annotators, system scores can be computed. The first step is to average any instances of scores that share the same system ID, the same document ID, and the same sentence ID (regardless of whether they are annotated by the same or different annotators). Then, all segments produced by a particular system are averaged into the final system score. These last two steps are performed on both raw scores and z-scores, but the ranking is computed using z-scores. Clusters of systems are indicated by horizontal lines in the ranking (Tables 3 and 4), with such a horizontal line drawn below a system if and only if its z-scores are significantly better than all systems ranked below it according to a Wilcoxon ranked sum test ($p < 0.05$). The differences between this and the standard WMT data collection are the choice to compute z-scores over annotators rather than over annotator sessions and the fact that we did not collect any "BAD" quality assurance annotations.

3.2 News Ranking Approach

Ave.	Ave.z	System	Findings Ranking
90.3	0.652	Human-A	(1-2, 90.5, 0.574)
76.4	0.219	CUNI-Transfer	(3-9, 77.4, 0.409)
77.7	0.102	NICT_Kyoto	(3-9, 79.2, 0.364)
71.6	0.096	NRC	(3-9, 71.9 0.369)
76.2	0.053	Ubiqus	(1-2, 75.3, 0.425)
74.1	0.041	Helsinki	(3-9, 75.2, 0.296)
73.6	0.025	Facebook_AI	(3-9, 74.6, 0.368)
72.7	0.012	SRPOL	(3-9, 72.8, 0.282)
72.8	-0.052	Groningen	(3-9, 71.6, 0.339)
67.6	-0.305	UQAM_TanLe	(10-11, 68.9, 0.084)
65.0	-0.427	UEDIN	(10-11, 66.4, 0.081)
46.8	-1.223	OPPO	(12, 48.2, -0.384)
0.0	-3.181	zlabs-nlp	(not shown)

Table 4: News Rankings, with mean and standard deviation for z-score computed using only SRPOL (all annotators scored output from that system). The last column shows the systems' original rankings in the 2020 Findings paper: cluster range, raw average, and z-average.

For the News task (Table 4), we had the N-1 and N-2 datasets along with a mapping between annotators and annotation sessions. Starting from this, we modified the ranking computation process to attempt to account for the known concerns with the dataset. We were unable to replicate the Findings rankings (Barrault et al., 2020), nor the listed number of annotations for Inuktitut from the data released.[9] The Findings rankings may have been

[9] https://www.statmt.org/wmt20/results.html

computed from earlier, incomplete data.

Due to the high average document length (39.0 lines) and the extreme range of system quality (from human reference near 100 to zlabs at 0), we cannot expect annotation sessions to be comparable to one another and certainly not representative of the whole test data and systems. For this reason, it is already not appropriate to compute z-scores at the annotation session level. Additionally, it is not appropriate to compare z-scores that are computed in the standard way between the N-1 and N-2 datasets, since the former does not contain human references while the latter does. Adding to the challenges, not all annotators completed annotation sessions in both of the datasets, and not all annotators annotated data from all systems (or across systems in the same proportions). Thus, simply switching to the annotator-level z-score computation does not solve the problem. For this reason, we chose to compute the mean and standard deviation for each annotator based only on the SRPOL system segments that they had annotated. SRPOL and CUNI were at the intersection of all annotators' sets of annotated systems, but in different ratios, so we selected SRPOL because the annotator who had annotated the smallest number of segments had annotated more SRPOL than CUNI segments. We do not include "BAD" segments in the z-score calculations (as different annotators had different proportions of quality assurance data) and we also do not eliminate any data based on quality assurance measures. This does not guarantee that this is a perfectly fair comparison, as the specific documents and segments annotated are not consistent across annotators, but it does limit the influence of extreme outliers on the z-score computations. We then use those means and standard deviations to compute z-scores for all data across all systems.

In conjunction with these justifications, we note the following as additional support for our chosen approach to ranking the News data. The stated goal of using z-scores (rather than raw scores) in the official ranking is "to iron out differences in scoring strategies of distinct human assessors" (Barrault et al., 2020). If we had perfectly consistent annotators and were computing z-scores in such a way that they were standardizing annotator difference rather than other information in the data, z-scores and raw scores would produce matching orderings of systems. If all annotators were perfectly consistent but the z-scores did not correlate with the raw scores, then we would know that there was a problem with the z-score calculations or annotation setup. We simulate this by replacing all News human annotation scores with CHRF scores as pseudo-annotations and then calculate rankings in approximately the style of WMT20 by computing z-scores at the annotation session level[10] and then computing them using our approach. We find z-scores and raw scores produce identical system orderings under our approach, but produce less-correlated (i.e., non-identical) orderings using the WMT20 approach. Computing means and standard deviations for CHRF scores at the annotator level (but across all systems) does improve the most extreme differences between raw and z-scores, but the complete ordering is still not as well-correlated as with our new approach. While this does not guarantee that our approach fully solves the problem, it does demonstrate that our approach does not introduce the same error as the WMT20 approach.

If we were to use only SRPOL data for computing the annotator means and standard deviations for the Hansard ranking, we would obtain the same ordering of systems that we obtained via the approach described in Section 3.1 (though of course with different z-scores), with the only difference being that using SRPOL only would put UEdin and Helsinki in the same significance cluster.

4 Rankings

We observe several similarities and differences between the Hansard (Table 3) and News (Table 4) rankings. As expected, the authentic Human translations consistently score highly (with raw scores of 90.3 for News and 88.1 for Hansard) and are in the top cluster of the rankings. In the case of News, the authentic human translations are in a cluster of their own, while in the case of the Hansard data, they are in a cluster alongside the four top-performing MT systems. The raw scores for the News rankings are consistently lower than those for the Hansard rankings (both overall and on a system-by-system basis). This reflects the fact that there is less data in the News domain and the fact that the Hansard domain is highly repetitive. We will explore both of these topics in Section 5.3.

[10] This is intended to be closer to what we believe was done at WMT20; however, the WMT20 calculation for means and standard deviations likely included "BAD" reference segments, which we must omit because we do not have the "BAD" reference text to be able to compute CHRF scores against the reference.

The system that shows the smallest gap in raw scores between Hansard and News and the greatest improvement in clustering, moving from the fifth cluster for Hansard to the second for News was Ubiqus, which saw a difference of just 1.8. In comparison, the systems with the greatest drops in rankings (UEdin from third cluster to sixth, and Groningen from one to four) saw raw score average drops between 14.7 and 20.6. The OPPO and NRC systems also saw large drops in raw average scores (18.8 and 17.2, respectively) but with smaller or no corresponding ranking drops (in the case of OPPO, it was ranked last in Hansard so no drop was possible).

5 Discussion

5.1 System Performance

Here we discuss system performance across the different test sets. Table 5 summarizes some of the features of the approaches used in different submissions, while Figure 1 visualizes our two rankings and the published Findings ranking from Barrault et al. (2020). All submitted systems for which we have information used Transformer models, implemented in a range of toolkits.

System	Toolkit	BT	Tag	News Dev
CUNI	tensor2tensor	Y	-	-
Facebook	fairseq	Y	Y	75%
Groningen	Marian	Y	Y	76%
Helsinki	OpenNMT-py	Y	-	-
NICT				
NRC	Sockeye	Y	Y	100%
OPPO	fairseq	Y	-	-
SRPOL	Marian	Y	-	-
Ubiqus	OpenNMT-py	Y	-	-
UEdin	Marian	Y	-	-
UQAM				
zlabs				

Table 5: Table summarizing system features (where known), including toolkit used, use of backtranslation (BT), use of tags for domain and/or backtranslation (Tag), and whether News development data was used in training. For systems without a corresponding system description, unknown information is left blank. Unconstrained systems are marked in grey.

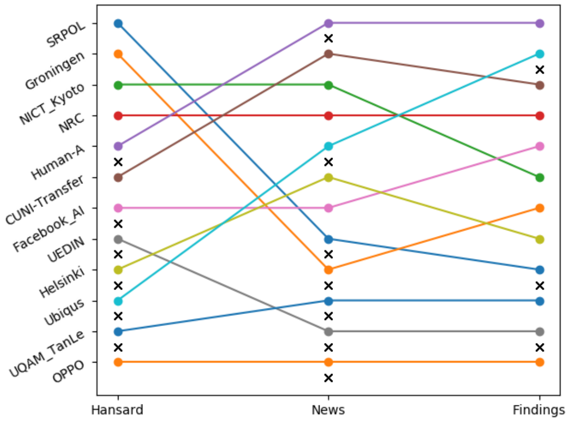


Figure 1: Summary of differences in clusterings and rankings. Black x marks indicate the demarcation between clusters and the systems are listed from best performing (top) to worst performance (bottom) across our Hansard ranking, our News ranking, and the ranking from the Findings paper (which used only News data).

Two systems participated as "unconstrained" systems (incorporating additional data), with differing levels of success. Multilingual_Engine_Ubiqus moves from the bottom half of the systems when ranked on Hansard to the top half when ranked on News, and their system incorporated additional data from news and magazine domains. This may account for some of the performance improvement they observed, though we cannot say with certainty if this is the main or only factor. On the other end, UQAM's system was also unconstrained but did not perform as well on News data. While examining the annotation data, we observed that despite its relatively low ranking, the UQAM system had a high number of segments with perfect automatic metric scores (CHRF of 100.0), meaning they were identical to the reference. Upon closer inspection, we found that 24.6% (295 of 1201) of UQAM segments annotated and labeled TGT (target) in the human annotation were identical to the reference. This compares to just 1.2% (183 of 14772) of all other systems' TGT annotated segments (excluding Human, which is itself the reference). The UQAM segments that were identical to the reference received very high scores, in line with the general trend for human translations. While there was not a paper submitted with the UQAM submission, the fact that it was marked as unconstrained suggests that their approach included additional data collection, and it appears that this included some of the test data.

All systems that had corresponding papers incorporated backtranslation. Three systems incorporated training on News development data (in various quantities) and these same three used tags to indicate domain and backtranslation. While this may have benefited the systems that incorporated it, it's clear that it was neither necessary nor sufficient to guarantee that a system placed in the top cluster.

Other techniques that systems used included pretraining, monolingual tasks, BPE dropout, ensembling, transfer learning, and more. There remains work to be done to identify which approaches produce the most positive impacts on translation quality. As it stands, a major challenge is that the development of the systems relied on automatic metrics, without knowing for sure which automatic metrics might be best suited to this language pair (several papers note the use of character-level metrics due to their prior results on morphologically complex languages). In Section 6 we will discuss the correlation between automatic metric scores and the human annotation results, in the hopes that this will be useful for future work on this language pair.

In addition to the rankings, we take a closer look at the performance of systems in Figures 2 and 3, which show raw human annotations averaged by document. These provide a rough visualization of system performance across different documents, as well as highlighting differences between the domains. The visualization shows both the lower overall scores assigned to the News data, as well as the greater coverage of the annotations in the Hansard data. We also see that certain documents are easy for most systems to translate, while others are consistently more difficult across systems. For example, the two documents with the highest median segment level scores, Hansard sub-documents 106 and 107, both consist of lists of names and positions, as well as standard parliamentary text about the house adjourning. Those documents with the lowest median segment scores contain longer sentences of members' speeches across varied topics like the Indspire awards, Red Seal program trades, and so on. We can also see how some systems perform relatively consistently across documents, while others exhibit more anomalous behaviour. For example we can see that the UQAM system exhibits some extremes (and the high-scored News document from 2019-11-12 is one where we note that the system output is identical to the reference, likely due to the system being unconstrained).[11]

5.2 Annotation Data Coverage

For the Hansard data collection, all systems were annotated over at least 97% of test segments, whereas for News data, coverage ranged from 47%

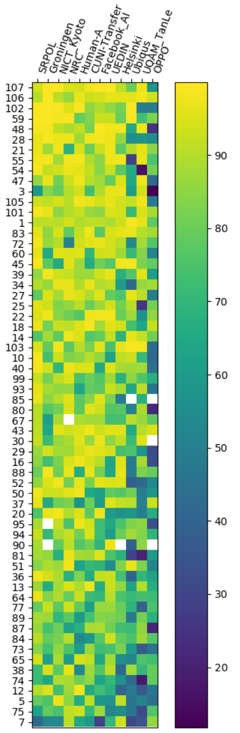


Figure 2: Raw annotation scores, averaged per document, for Hansard data. Lighter/brighter colors indicate higher scores; systems are ordered according to the rankings, while documents are ordered according to median segment score across all systems for that document. Blank spaces indicate no annotation for that document-system pair.

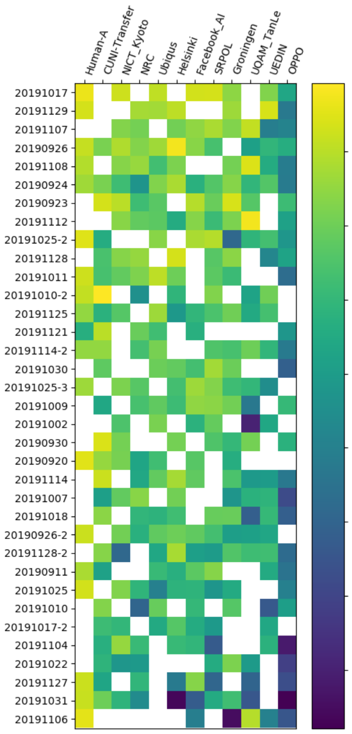


Figure 3: Raw annotation scores, averaged per document, for News data.

[11] The other systems that do see particularly high-performing documents do not have those as exact matches to the references, and in one case it is likely due to the fact that the news article is about a bill in the Legislative Assembly.

to 73% of test segments (with the human reference translations the least annotated).[12] This means that most Hansard "documents" were annotated for most systems (101 out of 107 had annotations for every system), while there were no News documents (out of 36) annotated for all systems. As we observe that some documents may be easier or more difficult for most systems, annotating all systems over nearly the same set of documents aims to alleviate this potential source of error.

5.3 Repetition and Novelty

Of the approximately 1.3 million sentence pairs in the Hansard training data, 59.8% of these are unique pairs, while the remainder are duplicates. The 2971 lines of the test set are all unique, and of these 15 sentence pairs were observed in the Hansard training data. If we consider only the source side, there are 155 source (English) sentences that appear in the test set that also appeared in the English side of the training data. Even though the target side of 140 of these segments is not identical to the target reference in the test set, systems still performed better on these previously observed segments than they did on segments that were previously unobserved. In fact, for all but the three lowest-performing systems, the raw scores on segments where the source had been observed in training averaged over 90.

In addition to the exact matches, the Hansard contains much boilerplate text, with small differences between what has been observed in training and the data in the test set. This includes segments like those at the start of a session, that indicate the date and time, as well as formulaic parliamentary speech (such as addressing the Speaker). All in all, the Hansard test data is more similar to the Hansard training data than the News test data is to the Hansard training data. Within each domain, there is not a strong correlation between source side similarity to training data and raw direct assessment scores, but across domains this may contribute to the differences we observe. This is also an imperfect analysis, as some systems used additional data and some incorporated development News data into their training. Nevertheless, we expect that the domain differences, compounded by the difference in data sizes, explain much of the difference in raw scores between the two domains.

6 Automated MT evaluation

Another important area of research on English–Inuktitut machine translation is accurate automated MT evaluation metrics for a polysynthetic language. Language model based metrics usually correlate better with human judgments when evaluating translation in non-polysynthetic languages but they suffer from a training resource scarcity problem when evaluating polysynthetic languages. Character based metrics are more commonly used for evaluating translation in low resource and polysynthetic languages (Mager et al., 2021) but there is not enough study on their correlation with human judgments. A complete collection of human annotations on both domains of the English–Inuktitut test set with translation output from diverse MT systems enables further studies on automated MT evaluation metrics, with the caveat that caution should be taken with News, due to the issues in the data collection described above and in Appendix B.

6.1 Setup

We rerun the correlation analysis of the WMT20 Metrics shared task (Mathur et al., 2020b) at system level and segment level with the updated system rankings on News and the newly-collected annotations on the Hansard data. Following the Metrics shared task setup, we use `mt-metrics-eval`[13] to conduct the correlation analysis.

The correlation analysis includes all the systems, except Human-A and zlabs-nlp. Human-A is excluded because a second reference was not available for the reference based metrics to score against and zlabs-nlp is excluded because this system was not included in the WMT20 Metrics shared task test set and thus none of the participants provided scores for it.

Following the official results in WMT20 Metrics shared task, we use Pearson's coefficient to examine system level correlations of metrics with and without outlier systems. The outlier systems[14] (Mathur et al., 2020a) for the News Rankings are OPPO, UEDIN and UQAM_TanLe while those for the Hansard Rankings are OPPO and UQAM_TanLe. It is important to note that for both domains, the outlier systems are all on the lower quality side.

[12] Adding in the Crowd annotations does not increase coverage, it simply increases the number of annotations for the sentences already annotated.

[13] https://github.com/google-research/mt-metrics-eval

[14] Systems that are greater than 2.5 median average deviation from the median.

	Human annotations	Findings News		News		Hansard	
	Metrics \ Systems	**all**	**all-out**	**all**	**all-out**	**all**	**all-out**
Character	characTER	0.515 (14)	0.121 (13)	0.504 (15)	-0.358 (20)	0.491 (11)	0.844 (2)
	chrF	0.336 (19)	0.091 (18)	0.355 (19)	-0.339 (17)	0.398 (14)	0.557 (12)
	chrF++	0.315 (20)	0.098 (15)	0.326 (20)	-0.323 (16)	0.344 (15)	0.566 (11)
	EED	0.483 (16)	0.122 (12)	0.495 (16)	-0.290 (15)	0.472 (12)	0.738 (6)
	YiSi-0	0.505 (15)	0.095 (16)	0.511 (14)	-0.346 (18)	0.451 (13)	0.784 (3)
Word	parbleu	0.126 (22)	0.306 (4)	0.181 (22)	-0.022 (5)	0.146 (20)	0.352 (21)
	sentBLEU	0.075 (23)	0.172 (8)	0.128 (23)	-0.152 (9)	0.048 (22)	0.503 (16)
	TER	0.357 (18)	0.083 (20)	0.441 (17)	-0.225 (12)	0.238 (18)	-0.106 (23)
Pretrn. LM	BLEURT-extended	0.762 (9)	0.155 (10)	0.759 (9)	-0.350 (19)	0.794 (7)	0.406 (19)
	COMET	0.858 (6)	0.152 (11)	0.853 (6)	-0.384 (23)	0.839 (2)	0.615 (9)
	COMET-2R	0.867 (4)	0.177 (7)	0.875 (4)	-0.152 (9)	0.725 (9)	0.735 (7)
	COMET-HTER	0.888 (3)	0.092 (17)	0.896 (3)	-0.228 (13)	0.818 (4)	0.355 (20)
	COMET-MQM	0.867 (4)	0.172 (8)	0.854 (5)	-0.368 (21)	0.825 (3)	0.463 (17)
	COMET-Rank	0.392 (17)	0.252 (5)	0.420 (18)	-0.061 (6)	0.069 (21)	0.651 (8)
	MEE	0.242 (21)	0.113 (14)	0.260 (21)	-0.285 (14)	0.219 (19)	0.579 (10)
Custom LM	YiSi-1	0.523 (13)	-0.014 (22)	0.529 (13)	-0.377 (22)	0.584 (10)	**0.852** (1)
Others	esim	0.760 (10)	0.418 (2)	0.740 (11)	-0.148 (7)	0.818 (4)	0.547 (13)
	paresim	0.760 (10)	0.418 (2)	0.740 (11)	-0.148 (7)	0.818 (4)	0.547 (13)
	prism	**0.945** (1)	0.088 (19)	**0.960** (1)	0.140 (3)	**0.974** (1)	0.775 (4)
Ref.-less	COMET-QE	0.928 (2)	**0.651** (1)	0.934 (2)	**0.534** (1)	0.298 (16)	0.237 (22)
	OpenKiwi-Bert	0.808 (8)	0.194 (6)	0.826 (8)	0.285 (2)	-0.170 (23)	0.455 (18)
	OpenKiwi-XLMR	0.680 (12)	-0.358 (23)	0.748 (10)	0.022 (4)	0.280 (17)	0.741 (5)
	YiSi-2	0.830 (7)	0.065 (21)	0.840 (7)	-0.217 (11)	0.746 (8)	0.540 (15)

Table 6: System-level Pearson's correlation of WMT20 Metrics shared task participants with z-score reported in WMT20, table 4 and 3. For WMT20 News and table 4 rankings, the outlier systems are UQAM_TanLe, UEdin and OPPO. For Hansard, the outlier systems are UQAM_TanLe and OPPO.

6.2 System-level correlation

Table 6 shows system-level Pearson's correlations of metrics with revised rankings on the News domain and new rankings on the Hansard domain.

When the outliers are included, the system-level correlations with the revised rankings are similar to those reported in the WMT20 Metrics shared task, based on the Findings rankings. However, we have several striking observations on the correlation with the revised rankings excluding the outlier systems:

- Most metrics show negative correlations with the revised rankings on the News domain.
- Reference-less metrics correlate better with revised rankings than reference based ones do.
- The rankings of the automated metrics change drastically when comparing against those obtained by correlating with the rankings in Barrault et al. (2020).
  - prism (Thompson and Post, 2020) and OpenKiwi-XLMR (Kepler et al., 2019) change from having the lowest correlation with the Findings rankings to being some of the very few metrics with a positive correlation with the revised rankings.
  - Similar changes can also be observed in YiSi-2 (Lo and Larkin, 2020) and TER (Snover et al., 2006) where they change from having the lowest correlation to the middle of the pack.
  - On the contrary, BLEURT-extended (Sellam et al., 2020), COMET (Rei et al., 2020), COMET-MQM and characTER (Wang et al., 2016) demote from the middle of the pack to having the lowest correlation with the revised rankings.

As we have established in section 3.2 that we believe the revised News rankings to be more accurate, the negative correlations with human achieved by the majority of the metrics reflect the difficulty in evaluating translation quality of low-resource polysynthetic languages for out-of-domain settings. It is important to note that the range of z-scores in the revised News rankings is [-0.052, 0.219]. It is a noticeably smaller range as compared against the range of z-scores in the Hansard rankings, which is [-0.127, 0.249]. The small variation of MT system performance in the News domain also increases the difficulty of the automated evaluation task.

Prism performs consistently well across domains with and without outliers. This is perhaps because it is one of the very few metrics that used the constrained English–Inuktitut data to train their metrics to evaluate translation quality in Inuktitut.

For the Hansard domain, it is not surprising to see YiSi-1 (Lo, 2020) correlating very well with hu-

	Metrics \ Annotations	**Findings News**	**News**	**Hansard**	**News+Hansard**
Character	characTER	0.309 (11)	0.333 (11)	0.265 (6)	0.289 (6)
	chrF	0.344 (5)	0.373 (5)	0.293 (2)	0.321 (2)
	chrF++	0.338 (6)	0.368 (6)	0.288 (3)	0.317 (4)
	EED	0.361 (3)	0.395 (3)	0.277 (4)	0.319 (3)
	YiSi-0	0.362 (2)	0.396 (2)	0.268 (5)	0.313 (5)
Word	parbleu	0.212 (14)	0.232 (15)	-0.043 (19)	0.054 (18)
	sentBLEU	0.206 (15)	0.233 (14)	-0.004 (18)	0.080 (15)
	TER	-0.071 (21)	-0.051 (21)	-0.284 (23)	-0.201 (23)
Pretrained LM	BLEURT-extended	0.359 (4)	0.387 (4)	0.226 (7)	0.283 (7)
	COMET	0.322 (9)	0.342 (9)	0.147 (11)	0.216 (9)
	COMET-2R	0.326 (8)	0.344 (8)	0.143 (12)	0.214 (11)
	COMET-HTER	0.331 (7)	0.348 (7)	0.135 (13)	0.211 (12)
	COMET-MQM	0.313 (10)	0.337 (10)	0.127 (14)	0.202 (13)
	COMET-Rank	0.297 (12)	0.312 (12)	0.174 (10)	0.223 (8)
	MEE	-0.074 (22)	-0.054 (22)	-0.212 (22)	-0.156 (22)
Custom LM	YiSi-1	0.251 (13)	0.269 (13)	0.186 (9)	0.215 (10)
Others	esim	0.122 (17)	0.142 (17)	0.039 (15)	0.075 (16)
	paresim	0.122 (17)	0.142 (17)	0.039 (15)	0.075 (16)
	prism	**0.452** (1)	**0.475** (1)	**0.326** (1)	**0.379** (1)
Reference-less	COMET-QE	-0.040 (20)	-0.036 (20)	-0.084 (20)	-0.067 (20)
	OpenKiwi-Bert	-0.115 (23)	-0.098 (23)	-0.169 (21)	-0.143 (21)
	OpenKiwi-XLMR	0.060 (19)	0.062 (19)	0.036 (17)	0.045 (19)
	YiSi-2	0.146 (16)	0.147 (16)	0.189 (8)	0.174 (14)

Table 7: Segment-level Kendall's correlation of WMT20 Metrics shared task participants with raw scores collected in News (N-1 and N-2), Hansard (H-A and H-B) and News+Hansard.

mans when the outliers are excluded. It is because YiSi-1 is based on XLM (Lample and Conneau, 2019) trained on the constrained English–Inuktitut parallel training data in Hansard domain. Another observation is that for evaluating in-domain systems, character-based metrics, characTER and YiSi-0, correlate very well with humans. This is the first scientific evidence that character-based MT evaluation metrics are a better choice for evaluating translation quality in low-resource polysynthetic languages.

6.3 Segment-level correlation

Table 7 shows the segment-level Kendall's correlations of metrics. We observe much more consistency in metrics' correlation with humans at segment level than that at system level across domains. This is possibly due to the fact that there are more data points used for correlation analysis at the segment level than the system level. Similar to correlations at system level, prism consistently correlates the best with humans at segment level.

We see even stronger evidence here at segment level that all character-based metrics (Wang et al., 2016; Popović, 2015, 2017; Stanchev et al., 2019; Lo, 2019) correlate very well with humans for evaluating translation quality in polysynthetic languages across domains. This is a particularly important finding because these character-based metrics are resource-free. That means we now have strong confidence in using character-based metrics for evaluating translation quality in a low-resource polysynthetic language.

7 Conclusion

In this work we present additional human annotations for the Hansard portion of the WMT 2020 English–Inuktitut machine translation shared task test set. We provide new system rankings on this portion of the data and present revised rankings on the News portion. We demonstrate that these changes in rankings have downstream effects on the evaluation of automatic metrics for the shared task, and examine the difficulty of performing automatic evaluation on out-of-domain text in a polysynthetic language. When it comes to automatic metrics, we find that the top-performing system incorporated training data in the low-resource target language. However, character-level automatic metrics (which did not require training) also performed amongst the top systems, demonstrating their appropriateness for evaluating translation into Inuktitut. While additional research will be required to confirm that this finding generalizes to other polysynthetic languages, we release this expanded dataset to enable more study of automatic metrics for low-resource and polysynthetic languages.

Acknowledgements

We thank the language experts and our contacts at Pirurvik Centre for their work on the annotation tasks. We thank Roman Grundkiewicz, Tom Kocmi, and Christian Federmann for their assistance in preparing and hosting the second round of Appraise annotations. We thank Eric Joanis for his work on organizing the WMT20 task, preparing the data for the task and annotation, providing information about task details, and for his feedback. We thank Roland Kuhn for his role in organizing the shared task and for his feedback, and Darlene Stewart and Samuel Larkin for work during the WMT20 shared task that informed this work. We thank Gabriel Bernier-Colborne, Cyril Goutte, Michel Simard, Yunli Wang, Patrick Littell, our colleagues, and the anonymous reviewers for their suggestions and comments.

References

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Rachel Bawden, Alexandra Birch, Radina Dobreva, Arturo Oncevay, Antonio Valerio Miceli Barone, and Philip Williams. 2020. The University of Edinburgh's English-Tamil and English-Inuktitut submissions to the WMT20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 92–99, Online. Association for Computational Linguistics.

Peng-Jen Chen, Ann Lee, Changhan Wang, Naman Goyal, Angela Fan, Mary Williamson, and Jiatao Gu. 2020. Facebook AI's WMT20 news translation task submission. In *Proceedings of the Fifth Conference on Machine Translation*, pages 113–125, Online. Association for Computational Linguistics.

Benoît Farley. 2009. The Uqailaut Project.

Yvette Graham, Timothy Baldwin, Meghan Dowling, Maria Eskevich, Teresa Lynn, and Lamia Tounsi. 2016. Is all that glitters in machine translation quality estimation really gold? In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3124–3134, Osaka, Japan. The COLING 2016 Organizing Committee.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2014. Is machine translation getting better over time? In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 443–451, Gothenburg, Sweden. Association for Computational Linguistics.

François Hernandez and Vincent Nguyen. 2020. The ubiqus English-Inuktitut system for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 213–217, Online. Association for Computational Linguistics.

Eric Joanis, Rebecca Knowles, Roland Kuhn, Samuel Larkin, Patrick Littell, Chi-kiu Lo, Darlene Stewart, and Jeffrey Micher. 2020. The Nunavut hansard Inuktitut–English parallel corpus 3.0 with preliminary machine translation results. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2562–2572, Marseille, France. European Language Resources Association.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Rebecca Knowles. 2021. On the stability of system rankings at WMT. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Rebecca Knowles, Darlene Stewart, Samuel Larkin, and Patrick Littell. 2020. NRC systems for the 2020 Inuktitut-English news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 156–170, Online. Association for Computational Linguistics.

Tom Kocmi. 2020. CUNI submission for the Inuktitut language in WMT news 2020. In *Proceedings of the Fifth Conference on Machine Translation*, pages 171–174, Online. Association for Computational Linguistics.

Mateusz Krubiński, Marcin Chochowski, Bartłomiej Boczek, Mikołaj Koszowski, Adam Dobrowolski, Marcin Szymański, and Paweł Przybysz. 2020. Samsung R&D institute Poland submission to WMT20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 181–190, Online. Association for Computational Linguistics.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

Tan Ngoc Le and Fatiha Sadat. 2020. Low-resource NMT: an empirical study on the effect of rich morphological word segmentation on Inuktitut. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 165–172, Virtual. Association for Machine Translation in the Americas.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Chi-kiu Lo. 2020. Extended study on using pretrained language models and YiSi-1 for machine translation evaluation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 895–902, Online. Association for Computational Linguistics.

Chi-kiu Lo and Samuel Larkin. 2020. Machine translation reference-less evaluation using YiSi-2 with bilingual mappings of massive multilingual language model. In *Proceedings of the Fifth Conference on Machine Translation*, pages 903–910, Online. Association for Computational Linguistics.

Manuel Mager, Arturo Oncevay, Abteen Ebrahimi, John Ortega, Annette Rios, Angela Fan, Ximena Gutierrez-Vasques, Luis Chiruzzo, Gustavo Giménez-Lugo, Ricardo Ramos, Ivan Vladimir Meza Ruiz, Rolando Coto-Solano, Alexis Palmer, Elisabeth Mager-Hois, Vishrav Chaudhary, Graham Neubig, Ngoc Thang Vu, and Katharina Kann. 2021. Findings of the AmericasNLP 2021 shared task on open machine translation for indigenous languages of the Americas. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 202–217, Online. Association for Computational Linguistics.

Benjamin Marie, Raphael Rubino, and Atsushi Fujita. 2020. Combination of neural machine translation systems at WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 230–238, Online. Association for Computational Linguistics.

Joel Martin, Howard Johnson, Benoit Farley, and Anna Maclachlan. 2003. Aligning and using an English-Inuktitut parallel corpus. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pages 115–118.

Joel Martin, Rada Mihalcea, and Ted Pedersen. 2005. Word alignment for languages with scarce resources. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 65–74, Ann Arbor, Michigan. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020a. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020b. Results of the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Jeffrey Micher. 2017. Improving coverage of an Inuktitut morphological analyzer using a segmental recurrent neural network. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 101–106, Honolulu. Association for Computational Linguistics.

Jeffrey Micher. 2018. Using the Nunavut hansard data for experiments in morphological analysis and machine translation. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, pages 65–72, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram f-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Maja Popović. 2016. chrF deconstructed: beta parameters and n-gram weights. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 499–504, Berlin, Germany. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Unbabel's participation in the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 911–920, Online. Association for Computational Linguistics.

Christian Roest, Lukas Edman, Gosse Minnema, Kevin Kelly, Jennifer Spenader, and Antonio Toral. 2020. Machine translation for English–Inuktitut with segmentation, data acquisition and pre-training. In *Proceedings of the Fifth Conference on Machine Translation*, pages 274–281, Online. Association for Computational Linguistics.

Yves Scherrer, Stig-Arne Grönroos, and Sami Virpioja. 2020. The University of Helsinki and aalto university submissions to the WMT 2020 news and low-resource translation tasks. In *Proceedings of* *the Fifth Conference on Machine Translation*, pages 1129–1138, Online. Association for Computational Linguistics.

Lane Schwartz, Francis M. Tyers, Lori S. Levin, Christo Kirov, Patrick Littell, Chi-kiu Lo, Emily Prud'hommeaux, Hyunji Hayley Park, Kenneth Steimel, Rebecca Knowles, Jeffrey Micher, Lonny Strunk, Han Liu, Coleman Haley, Katherine J. Zhang, Robbie Jimerson, Vasilisa Andriyanets, Aldrian Obaja Muis, Naoki Otani, Jong Hyuk Park, and Zhisong Zhang. 2020. Neural polysynthetic language modelling. *CoRR*, abs/2005.05477.

Thibault Sellam, Amy Pu, Hyung Won Chung, Sebastian Gehrmann, Qijun Tan, Markus Freitag, Dipanjan Das, and Ankur Parikh. 2020. Learning to evaluate translation beyond English: BLEURT submissions to the WMT metrics 2020 shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 921–927, Online. Association for Computational Linguistics.

Tingxun Shi, Shiyu Zhao, Xiaopu Li, Xiaoxue Wang, Qian Zhang, Di Ai, Dawei Dang, Xue Zhengshan, and Jie Hao. 2020. OPPO's machine translation systems for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 282–292, Online. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Peter Stanchev, Weiyue Wang, and Hermann Ney. 2019. EED: Extended edit distance measure for machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 514–520, Florence, Italy. Association for Computational Linguistics.

Miloš Stanojević, Amir Kamran, Philipp Koehn, and Ondřej Bojar. 2015. Results of the WMT15 metrics shared task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 256–273, Lisbon, Portugal. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. CharacTer: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 505–510, Berlin, Germany. Association for Computational Linguistics.

A Context and Related Work

There is a dialect continuum of Inuit languages, including Inuktitut, that spans Arctic communities from Alaska to Greenland. The term Inuktut is often used to refer to parts of that dialect continuum, including Inuktitut.[15] There are two main orthographies used to write these languages: Roman orthography (Latin alphabet, *qaliujaaqpait*) and syllabics (*qaniujaaqpait*).[16] The language is morphologically complex – individual words are constructed of multiple morphemes – and a word may correspond to a whole phrase or more when translated into English.

There has been a range of computational work on Inuktitut over the past decades. This includes early work on alignment and the Nunavut Hansard (Martin et al., 2003, 2005) and the recent release of a new version of the aligned Nunavut Hansard, used as training data in this task (Joanis et al., 2020). Morphological analysis and segmentation have also been areas of interest (Farley, 2009; Micher, 2017). There is also prior work on machine translation (Micher, 2018; Schwartz et al., 2020; Joanis et al., 2020; Le and Sadat, 2020).

There has been limited to no work on human and automatic evaluation of machine translation into Inuktitut prior to this work. Prior work has shown that character-based automatic metrics demonstrate promising performance on morphologically rich languages, at least in part because they do not penalize morphological variation as much as word-level exact-match metrics do (Stanojević et al., 2015; Popović, 2016). Put another way, they award "partial credit" when a system produces some but not all of the morphemes of a word correctly. This is particularly important when translating into polysynthetic or morphologically complex languages. While our results in this paper show the promise of character-level metrics, it would be useful for future work to provide a more in-depth examination of their performance to better understand their success, perhaps with analysis at the word level and not simply the sentence level.

[15] https://tusaalanga.ca/about-Inuktut
[16] https://tusaalanga.ca/node/2505

B Quotation Marks

During the test set submission period at WMT20, it was noted that a number of segments in the test set were wrapped in ASCII quotes. This was specifically an issue with the News portion of the test set; 844 News segments exhibited this ASCII quote wrapping on source, target, or both, while just 561 of the News segments were unaffected by this. As the submission period was already underway, the task organizers made the decision not to change the test set and indicated that the annotators would be told not to take the quotation mark issues into account during their evaluation.

There remain, however, several ways that this problem may have impacted the task and its results. The first is that it may have altered the behavior of MT systems, as different systems may be more or less robust to this kind of variation in input. As we do not have access to most of the MT systems, we cannot test this. The second is that teams may have handled this differently, with some adding specialized preprocessing to deal with the wrapped quotation marks and others not, and not all system description papers indicate whether or not there was special handling of this issue. Lastly, it can have an effect on automatic metric behavior. We explore that briefly below.

If we examine just the set of segments with these spurious quotations on the target side, and compute BLEU using the segments with quotes as the reference, and identical segments but with the quotes removed as the hypothesis, we see the BLEU score drop more than 10 points (from a perfect score). Since there are so many segments with these quotation marks, we still see drop of more than 5 points when we expand to the full news portion of the test data. The impact on CHRF scores is smaller.

These spurious quotation marks, while not semantically meaningful, have varied impacts on automatic metric scores, and may have also had varied impacts on translation performance across MT systems. Unfortunately, because they make up such a large portion of the News portion of the test set, omitting them dramatically shrinks the pool of data available for computing rankings and correlations. Thus, we present this work with them included, and provide these caveats about the data.

C Quality Assurance

The quality control task used in out-of-English translation directions at WMT 2020 was "BAD reference pairs", which are segments where a short segment of a translation is randomly replaced with an equal length segment randomly selected from a different reference segment. For more details on their construction see (Barrault et al., 2020). The theory is that an annotator should score the "BAD" version of a segment lower than the original version of the same segment. If an annotator does not do so over the course of an annotation session, that session would be removed.

We note that there is a reason to not fully trust this particular approach to quality control for the News dataset. The system submitted under the name zlabs-nlp (no corresponding paper submitted) consistently received scores of 0 because it was identical to the English source. In most cases, the "BAD" references paired with zlabs-nlp segments also received scores of zero, but in a few cases they received low but non-zero scores. Unfortunately, because the text of the "BAD references" were not released by the organizers, we cannot examine this more closely, or determine whether this problem may also extend to other systems.

The quality assurance tasks typically used at WMT are included in order to exclude annotators' data from the final evaluation; in particular this would include annotators who are not adequately familiar with the language pair, who are not performing careful analyses, or who might be attempting to game a crowdsourcing task. While it may be easy to simply replace annotators for certain language pairs with very large bilingual populations, there is a much smaller number of fluent bilingual speakers of English and Inuktitut. This, combined with the very high demand for their language skills (e.g., in translation), meant that we chose to work with the Pirurvik Centre, who recruited a small number of highly-skilled fluent speakers to participate in this work. Thus, the annotators' language skills and quality of work (in different language-related tasks) are known to be high (unlike in a crowdsourcing scenario, where little information is typically known about participants).

In future work and under less tight constraints as regards annotator time and budget, we would encourage the collection of repeated annotation data. This could include repeated annotations performed by the same annotator (intra-annotator agreement) as well as repeated annotations across annotators (such as a calibration HIT that all annotators complete to examine inter-annotator agreement).

Continuous Rating as Reliable Human Evaluation of Simultaneous Speech Translation

Dávid Javorský **Dominik Macháček** **Ondřej Bojar**

Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
{surname}@ufal.mff.cuni.cz

Abstract

Simultaneous speech translation (SST) can be evaluated on simulated online events where human evaluators watch subtitled videos and continuously express their satisfaction by pressing buttons (so called Continuous Rating). Continuous Rating is easy to collect, but little is known about its reliability, or relation to comprehension of foreign language document by SST users. In this paper, we contrast Continuous Rating with factual questionnaires on judges with different levels of source language knowledge. Our results show that Continuous Rating is easy and reliable SST quality assessment if the judges have at least limited knowledge of the source language. Our study indicates users' preferences on subtitle layout and presentation style and, most importantly, provides a significant evidence that users with advanced source language knowledge prefer low latency over fewer re-translations.

1 Introduction

Simultaneous speech translation (SST) is a technology that assists users to understand and follow a speech in a foreign language in real-time. The users may need such an assistance because of limited knowledge of the source language, the speaker's non-native accent, or the topic and vocabulary. The technology can be used for the target languages, for which human interpretation is unavailable, e.g. due to capacity reasons.

Candidate systems for simultaneous speech translation differ in quality of translation, latency and the approach to stability. Some are streaming, only adding more words (Grissom II et al., 2014; Gu et al., 2017; Arivazhagan et al., 2019; Press and Smith, 2018; Xiong et al., 2019; Ma et al., 2019; Zheng et al., 2019; Iranzo Sanchez et al., 2022), some allow re-translation as more input arrives (Müller et al., 2016b; Niehues et al., 2016; Dessloch et al., 2018; Niehues et al., 2018; Arivazhagan et al., 2020). Finally, subtitle presentation options (size of subtitling window, layout, allowed reading time, font size, etc.) also affect users' impression. The combination of the re-translating approach and limited space for subtitles is challenging because of "flicker", i.e. the updates to the text that the user is reading at the moment, has already read, or that has been scrolled away. The subtitling options impact the amount of flicker, reading comfort and delay and may affect the general usability.

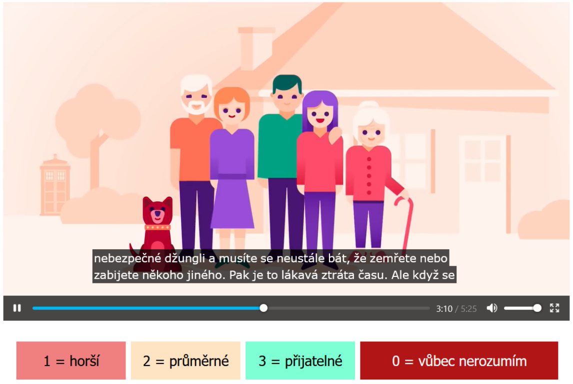

nebezpečné džungli a musíte se neustále bát, že zemřete nebo
zabijete někoho jiného. Pak je to lákavá ztráta času. Ale když se

Figure 1: A detail of the default layout with the video document "Dinge Erklärt: Impfen...".[1] The video is at the top, overlaid by two lines of subtitles in Czech, followed by buttons for Continuous Rating. The button labels are: 1: Worse; 2: Average; 3: Good; 0: I do not understand at all.

The evaluation of the traditional, text-to-text machine translation (MT) has been researched for many years (see e.g. Han, 2018 or developments and discussion within the series of WMT, Akhbardeh et al., 2021). It targets only the translation quality. SST evaluation faces new challenges: simultaneity, latency, and readability to humans. Evaluating only selected aspects in isolation is reasonable (as MT quality in Elbayad et al., 2020, latency in Ma et al., 2018; Cherry and Foster, 2019), however, a complete evaluation must be end-to-end, from sound acquisition to subtitling, and take into

[1] https://youtu.be/4E0dwFS72gk

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 154 - 164
December 7-8, 2022 ©2022 Association for Computational Linguistics

account the intent of communication. We generalize the intent to passing pieces of information from the speaker (sender) to a participant in an online session (receiver).

Our Contributions In this paper, we run an experimental evaluation campaign on 2 hours of documents with German-Czech SST using 32 judges with different levels of source language proficiency. (i) We contrast two methods of SST evaluation: Continuous Rating and factual questionnaires. We find out that Continuous Rating by bilinguals is easy and reliable for assessing the comprehension. (ii) We measure how much comprehension is lost by simultaneity, flicker and presentation options. (iii) We evaluate different presentation options and layouts and find the most preferred one. (iv) We find a statistically significant evidence that the users with an advanced, but limited knowledge of the source language reach higher comprehension with low latency subtitles than with large latency and low flicker. (v) We publish our implementation of the subtitling tool, web application for simulating live events with SST subtitling, and SST human evaluation framework.

Since Continuous Rating is easily applicable to any speech documents, even to those without transcripts and reference translations, and requires minimal time overhead for both preparation and user evaluation, we believe it is suitable to become a standardized way for human manual evaluation of SST.

2 Related Work

Hamon et al. (2009) propose user evaluation of speech-to-speech simultaneous translation. To test the adequacy and intelligibility, they prepared questionnaires with factual questions from the source speech. The judges listened either to the interpreter, or the machine, and answered the questions. They evaluated the offline mode, the judges were allowed to stop and replay the audio while answering. This way the authors measured the comprehension loss caused by the automatic translation or interpretation. Each sample was processed by multiple judges, to eliminate human errors. Fluency was assessed by the judges on a scale.

Macháček and Bojar (2020) propose a technique for collecting continuous user rating while the user watches video and simultaneous subtitles. The user is asked to express the satisfaction with the subtitles at any moment by pressing one of four buttons as the rating changes.

Müller et al. (2016a) analyzed the feedback from foreign students using KIT Lecture Translator within two semesters. Such a long-term and informal evaluation differs considerably from judging in controlled conditions. On one hand, it summarizes the real-life situation with all the variables and corner cases that a lab test could only approximate or omit. On the other hand, the users may not be motivated to give the feedback, and can give only personal opinions that may be biased. This way it is also difficult to compare multiple system candidates.

3 Evaluation Campaign

In our evaluation, we simulate live events on which participants need assistance with understanding the spoken language. The source and target languages in our study are German and Czech, respectively. This is an interesting example of two neighbouring countries, distinct language families and yet a relatively well studied pair with sufficient direct training data.

3.1 Translation System

We use the ASR system originally prepared for German lectures (Cho et al., 2013). It is a hybrid HMM-DNN model emitting partial hypotheses in real time and correcting them as more context is becoming available. The same system was used also by KIT Lecture Translator (Müller et al., 2016b).

The system is connected in a cascade with a tool for removing disfluencies and inserting punctuations (Cho et al., 2012), and with a German–Czech NMT system.

The machine translation is trained on 8M sentence pairs from Europarl and Open Subtitles (Koehn, 2005; Lison and Tiedemann, 2016), and validated on newstest. The Transformer-based (Vaswani et al., 2017) system runs in Marian (Junczys-Dowmunt et al., 2018) and reaches 18.8 cased BLEU on WMT newstest-2019.

Despite the translations are pre-recorded and only played back in our simulated setup, we ensured we keep the original timing as emitted by the online speech translation system.

3.2 Selection of Documents

We selected German videos or audio resources that fulfilled the following four conditions: 1) Length 5 to 10 minutes (with some exceptions). 2) The

Type	#	Length	Description
TP	3	18:08	European Parliament
TP	3	17:34	DG SCIC, Repository for interpretation training
A	3	27:52	A mock interpreted conference at interpretation school
V	2	14:43	Maus, Educative videos for children
A	2	18:48	DW, For learners of German
V	2	16:09	Dinge, Educative videos for teens
All	15	114:52	

Table 1: Summary of domains of selected documents. Type distinguishes audio only (A), talking person only (TP) and video (V) with illustrative or informative content. Length is reported in minutes and seconds.

translations had to be of a sufficient quality. Based on a manual check, we discarded several candidate documents: a math lecture and broadcast news due to many mistranslated technical terms and named entities. Another group of documents was mistranslated and discarded because they were not long-form speeches, but isolated utterances with long pauses. 3) Informative content. We intend to measure adequacy and comprehension by asking the judges complementary questions. We thus excluded the documents where the speaker is not giving information by speech, but uses mostly paralinguistic means, e.g. singing, poetry, or non-verbal communication. 4) Non-technicality. We expect the judges answer in several plain words in their mother tongue. They may lack knowledge of any specialized vocabulary.

We selected audios, videos with informative or illustrative content, and videos of talking persons, to compare user feedback for these types of documents. Table 1 summarizes the selected documents.

3.3 Subtitler: Subtitle Presentation

Subtitler is our implementation of the algorithm by Macháček and Bojar (2020) extended by automatic adaptive reading speed in addition to the "flicker" parameter as defined in Macháček and Bojar (2020). The speed varies between 10 and 25 characters per second depending on the current size of the incoming buffer. The default font size is 4.8 mm. The default subtitling window is 2 lines high and 163 mm wide.[2] By default, we use the maximum flicker and the lowest delay (presenting all translation hypotheses, not filtering out the partial and possibly unstable ones), no colour highlighting, and smooth slide-up animation while scrolling. The example of the setup can be seen in Figure 1.

With the default subtitling window, 90% of the words in the test documents are finalized in subtitles at most 3 seconds after translation. In 99%, it is at most 7 seconds. More details and the comparison to fixed reading speed are provided in Appendix A.1.[3]

3.4 Web Application as Simulation Environment

We implemented a web application for presenting video and audio documents with embedded Subtitler. We use it for simulation of live subtitled events. The application is equipped with a tool for collecting users' feedback. It also allows administrators to design experiments with different variables (document, subtitling layout, subtitling option) and distribute them to individual judges.[4]

3.5 Types of Feedback

Continuous Rating Inspired by Macháček and Bojar (2020), we add 4 buttons below the audio/video document. While watching, the participants are asked to press the buttons to indicate their current satisfaction with the subtitles. We let participants decide the frequency of rating but we suggest clicking each 5-10 seconds or when their assessment has changed. We encourage them to provide feedback as often as possible even if their assessment has not changed. The scores of the rating range between 0 (the worst) and 3 (the best). The order 1, 2, 3, 0 matches the keyboard layout; participants are encouraged to use keyboard shortcuts. The layout is illustrated in Figure 1.

Questionnaires Answering questions as an evaluation approach has been already used (Hamon et al., 2009; Berka et al., 2011). Our questionnaires were composed of two parts: factual questions and general questions.

For **factual questions** we used the open style, i.e. asking for a short response, instead of yes/no or multiple choice to exclude guessing. We asked a Czech teacher of German to prepare the questions and an answer key from the original German documents, regardless of the machine translation. The teacher wrote the questions in Czech, and was instructed to prepare one question from every 30

[2] All typographical properties follow https://bbc.github.io/subtitle-guidelines/

[3] The source code of Subtitler is available at https://github.com/ufal/subtitler

[4] The source code of the application is available at https://github.com/ufal/continuous-rating

	Layout Experiments							
CEFR	0	A1	A2	B1	B2	C1	C2	all
Count	5	5	1	2	1	-	-	14
	Flicker Experiments							
	Z	**Begin.**		**Advanced**				
CEFR	0	A1	A2	B1	B2	C1	C2	all
Count	3	1	3	-	2	8	1	18
All	8	6	4	2	3	8	1	32

Table 2: The judges by their German proficiency levels on CEFR scale and their assignment to experiments. In Flicker experiment, the distribution to groups: Zero level, Beginners, Advanced.

seconds of the stream and distribute them evenly, if possible. The questions had to be answerable only after listening to the document, and not from the general knowledge. The complexity of the questions was targeted on the level that an ordinary high-school student could answer after listening to the source document once, if the student would not have any obstacles in understanding German. To reduce the effect of limited memory, the judges had an option in the questionnaire to indicate they knew the answer but forgot it. Furthermore, they had to fill, from which source they knew the answer: from the subtitles, from the speech, from an image on the video, or from their previous knowledge.

Finally, we evaluated the factual questions manually against the key, rating them at three levels: correct, incorrect, and partially correct.

After the factual questions, all the questionnaires had a common part with **general questions** where we asked the judges on their impression of translation fluency, adequacy, stability and latency, overall quality, video watching comfort, and a summary comment.

3.6 Judges

We have conducted two groups of experiments, each with different and distinct groups of judges.

In Comprehension and Layout experiments (Sections 4.1 and 4.2), we examined distinct subtitling features. We selected 14 native Czech speakers as judges. Their self-reported knowledge of German had to be between zero and B2 on the CEFR[5] scale, to ensure they need some level of assistance with understanding German. We also ensured they do not have knowledge of any other language which could help them understanding German.

Type	w. avg±std	t-test
Offline+voting	0.81±0.11	
Offline	0.59±0.16	***
Online, without flicker	0.36±0.16	***
Online, flicker, top layout	0.33±0.13	
Online, flicker, least preferred	0.31±0.16	

Table 3: Comprehension scores on all documents and judges. The average weighted by number of questions in document. *** denote the statistically significant difference (p-value< 0.01) between the current and previous line.

For Flicker experiments (Section 4.3), we found other 18 native Czech speakers with an unrestricted German proficiency, to contrast their feedback and level of German. For further analyses, we divided them into three groups. For brevity further in the paper, we denote the judges with no proficiency of German as "Zero" level group, with proficiency between A1 and A2 as "Beginners", and the others as "Advanced". See summary of the judges in Table 2.

The judges were paid for participation in the study. Each judge spent in total 2 hours on watching and 3 hours on the questionnaires. They watched the videos at their homes on their own devices. They were asked to customize their screen resolution and eye-screen distance to suit their comfort.

4 Results

First, we analyzed the comprehension levels (Section 4.1) and presentation layouts (Section 4.2). Then, we selected the most preferred layout and used it for examining the impact of flicker on comprehension in Flicker experiments (Section 4.3).[6]

4.1 Comprehension Levels

In our study, we assume comprehension can be assessed as a proportion of correctly answered questions. We assume the following model: A person without any language barrier and with non-restricted access to the document during answering the questionnaire can answer all questions correctly. With a language barrier and offline MT (unlimited perusal of the document while answering), some information may be lost in machine translation. More information is lost with one-shot access to online machine translation because of forgetting and temporal inattention. Some more information may be

[5] Common European Framework of Reference for Languages

[6] The collected data are available at http://hdl.handle.net/11234/1-4913

lost because of flicker, and some more because of suboptimal subtitling layout.

Our results confirm the assumed hierarchy of comprehension levels. Moreover, we notice that even the judges with offline MT give inconsistent answers. Combining them and counting answers as correct if at least one judge is correct leads to higher scores. We explain it by insufficient attention.

Table 3 summarizes the results on all documents. We measured that on average, 81% of information was preserved by machine translation (Offline+voting, i.e. one of two judges answered correctly). A single judge could find 59% of information (Offline). In an oracle experiment without flicker, when the machine translation gives the final hypotheses with the timing of the partial ones (i.e. as if it knew the best translation of the upcoming sentence), a single judge could answer 36%. In real setup with flicker and the most preferred subtitling layout (Online, flicker, top layout), 33% information was found, and 31% with less preferred. The standard deviation is between 11 and 16%.

We found statistically significant difference (two-sided t-test) between offline MT with voting and without it, and between offline and online MT.

4.2 Layout Preference

We analyzed effects of distinct subtitling features by contrastive experiments differing only at one feature, see the paragraphs in this section. We distributed them randomly among the judges, regardless of their German skills. After watching each document, the judge fills the questionnaire.

In all cases, the results show a slight insignificant preference towards one variant of the feature in all three types of feedback that we collect: "Comprehension" is the proportion of correctly answered factual questions, "Averaged Continuous Rating" is an averaged feedback from button clicks, and "Final rating" summarizes the responses in the general section of questionnaires.

For visually informative videos, we separately report the scores of "Watching comfort" which we collected in the general section of questionnaires. Some judges provided also textual feedback, examples are in Appendix B.2.

Side vs Below For videos and videos with a talking person, we consider two locations for the subtitle window: on the left side of the video, or below. The side window can be high but narrow (17 lines of 60 mm width, to match the height of the video), while the window underneath is short and wide (2 lines of 163 mm width). The first is more comfortable for reading, the latter for watching the video.

The results are in Table 4 on the left. There is a preference for the layout "below" when the video is informative, and for "side" otherwise.

Below vs Overlay The subtitling window can be placed over the video, as in films, or below. In the first case, the subtitles possibly hide an informative image content, in the latter case, there is a larger distance between the image and the subtitles. The results on non-German speaking judges are insignificantly in favor of overlay, see the middle of Table 4.

Highlighting Flicker Status The underlying rewriting speech translation system distinguishes three levels of status for segments (automatically identified sentences): "Finalized" segments no longer change. "Completed" segments are sentences which received a punctuation mark. They can be changed by a new update and the prediction of the punctuation may also change or disappear. They usually flicker once in several seconds. "Expected" segments are incomplete sentences, to which new translated words are still appended. They flicker several times per second.

It is a user interface question if the status of the segments should be indicated by highlighting, or if this piece of information would be rather disturbing. We experimented only with colouring text background in large and medium subtitling window for audio-only documents.

Our experiments show that the judges prefer highlighting flicker status in the large window. For the medium window, this inclination is less clear, see Table 5.

Size of Subtitling Window The subtitling window can be of any size. If the window is short and narrow, there is a short gap between an image and subtitles, which simplifies focus switching. On the other hand, a small window contains short history, so the user can miss translation content if it disappears while paying attention to the video. A small window may also accidentally cause a long subtitling delay if the translation was updated in the scrolled-away part of text. In this situation, Subtitler has to "reset" the subtitles and repeat the part. With a large window, the distance between the growing end of the subtitles and the image is

		Side vs Below		Below vs Overlay		Size of subtitling window	
		Side	Below	Below	Overlay	2 l.×163mm	5 l.×200mm
Final rating	audio					10 1.80 ±0.87	8 **2.75 ±0.97**
	talking	5 **2.80 ±1.33**	7 2.43 ±1.05	9 2.33 ±1.05	9 **2.78 ±1.13**	9 2.33 ±1.05	5 **2.80 ±1.60**
	video	1 1.00 ±0.00	3 **1.67 ±0.94**	5 1.40 ±0.80	8 **2.38 ±0.86**	5 1.40 ±0.80	3 **2.33 ±0.47**
	sum, avg	6 **2.50 ±1.38**	10 2.20 ±1.08	14 2.00 ±1.07	17 **2.59 ±1.03**	24 1.92 ±1.00	16 **2.69 ±1.16**
Compre-hension	audio					10 0.25 ±0.15	8 **0.31 ±0.15**
	talking	5 **0.34 ±0.25**	7 0.28 ±0.27	9 0.29 ±0.25	9 **0.39 ±0.20**	9 0.29 ±0.25	5 **0.40 ±0.21**
	video	1 0.18 ±0.00	3 **0.36 ±0.04**	5 0.26 ±0.14	8 **0.37 ±0.11**	5 0.26 ±0.14	3 **0.28 ±0.05**
	sum, avg	6 **0.31 ±0.24**	10 0.30 ±0.23	14 0.28 ±0.21	17 **0.38 ±0.17**	24 0.26 ±0.19	16 **0.33 ±0.16**
Avg. Cont. Rating	audio					10 0.90 ±0.71	8 **1.66 ±0.95**
	talking	5 1.56 ±1.00	7 **1.78 ±0.35**	9 1.65 ±0.52	9 1.65 ±0.99	9 **1.65 ±0.52**	5 1.09 ±0.78
	video	1 0.23 ±0.00	3 **1.21 ±0.45**	5 1.11 ±0.50	8 **1.15 ±0.77**	5 1.11 ±0.50	3 **1.35 ±0.31**
	sum, avg	6 1.33 ±1.04	10 **1.64 ±0.45**	14 **1.47 ±0.57**	17 1.42 ±0.93	22 1.21 ±0.70	16 **1.42 ±0.85**
Watching comfort	talking	5 2.80 ±0.75	7 **3.33 ±0.75**	9 3.43 ±0.73	9 **4.11 ±0.74**	7 **3.43 ±0.73**	5 2.80 ±0.98
	video	1 2.00 ±0.00	3 **3.00 ±1.63**	5 2.20 ±1.60	8 **3.00 ±1.00**	5 2.20 ±1.60	3 **2.33 ±1.25**
	sum, avg	6 2.67 ±0.75	10 **3.22 ±1.13**	14 2.92 ±1.32	17 **3.59 ±1.03**	12 **2.92 ±1.32**	8 2.62 ±1.11

Table 4: Results of the contrastive experiments for Side vs Below, Below vs Overlay and Subtitling window size: 2 lines height × 163 mm width vs 5 lines height × 200 mm width. The three numbers in each row and cell are the number of experiments, average and standard deviation. The higher score, the better. Comprehension rate is between 0 and 1, average continuous rating is between 0 and 3, the others on a discrete scale 1 to 5. Higher score in each experiment is bolded. The last row of each section summarizes the scores across document types.

Highlighting	No	Yes	No	Yes	No	No
Size [lines,mm width]	18×250 ("Large")		5×200 ("Medium")		18×250	5×200
Final rating	14 2.93 ±0.80	13 **3.31 ±1.14**	2 2.50 ±0.50	1 **4.00 ±0.00**	11 **2.91 ±0.79**	8 2.75 ±0.97
Comprehension	14 0.25 ±0.15	13 **0.30 ±0.12**	2 **0.44 ±0.18**	1 0.39 ±0.00	11 0.23 ±0.14	8 **0.31 ±0.15**
Avg. Cont. Rating	14 1.32 ±0.82	13 **1.42 ±0.74**	2 **2.19 ±0.50**	1 2.12 ±0.00	11 1.50 ±0.79	8 **1.66 ±0.95**

Table 5: Results of highlighting experiments on audio documents and subtitling window size 5 lines × 200 mm vs 18 lines × 250 mm. Description of numbers as in Table 4.

larger. The content stays longer, but it is more complicated to find a place where the user stopped reading before the last focus switch.

Depending on spatial constraints, it is always recommended to use as large window as possible, especially for documents without visual information, where focus switching between an image and subtitles is not expected. We tested two pairs of sizes on the same documents. The results are in Table 4 on the right. As we expected, the window with 5 lines was rated insignificantly better than with 2 lines in most scales and setups, but the 2-line reached a higher average watching comfort (2.92) that the 5-line setup (2.62).

For an audio-only document, we also tested the large (18 lines) vs. medium (5 lines) window, observing users' reported preference for the large one but slightly higher comprehension and continuous feedback for the medium one, see the right part of Table 4.

4.3 Flicker Experiments

We assume that the user behaviour differs by knowledge of the source language. We hypothesize that the Zero group of users and Beginners read all the subtitles all the time and do not pay attention to the speech. They do not mind large latency, but demand high quality translation, and comfortable reading without flicker. On the other hand, the users with an advanced knowledge of the source language may listen to the speech, try to understand on their own, and look at the subtitles only occasionally, when they are temporarily uncertain or need assistance with an unfamiliar word. They need low latency, and do not mind slightly lower quality.

To empirically test our hypothesis, we prepared two realistic setups: With flicker, the subtitles are presented immediately as available, but with frequent rewriting which discomforts the reader. Without flicker, the translations are delayed until the SST system confirms they will not change, and that usually happens during uttering the next sentence. We selected two videos for this experiment and distributed these setups uniformly between all groups of judges.

The results of comprehension are in Table 6. It shows that Advanced users achieve higher compre-

	Zero level	**Beginners**	**Advanced**
flic.	27 **0.34 ±0.16**	33 **0.33 ±0.16**	91 **0.58 ±0.19**
no f.	29 0.30 ±0.15	38 0.31 ±0.12	81 0.49 ±0.20
	insignificant	insignificant	$p < 0.01$

Table 6: Comprehension scores on a setup with flicker and no flicker, as rated by judges with different source language proficiency. The three numbers in each row and cell are the number of samples, average and standard deviation. Higher scores bolded. The difference between setups within Advanced group is statistically significant with $p < 0.01$.

		χ^2**-test** p**-values**	
	Zero level	**Beginners**	**Advanced**
OK/OK-	0.24	$\mathbf{1.8 \cdot 10^{-5}}$	$\mathbf{5.6 \cdot 10^{-5}}$
unknown	0.033	$\mathbf{1.7 \cdot 10^{-4}}$	$\mathbf{9.1 \cdot 10^{-4}}$
wrong	0.59	0.45	$\mathbf{2.9 \cdot 10^{-3}}$
forgot	0.9	0.48	0.019

Table 7: The results of χ^2-test for independence of Continuous Rating and answer correctness. Bolded values are where the two variables are **dependent** with statistical significance $p < 0.01$.

hension with flicker (58%) than without (49%). We found the difference statistically significant, which confirms the second part of our hypothesis.

The Zero level speakers and Beginners also report higher comprehension with flicker (Zero: 30% vs 34% and Beginners: 31% vs 33%), but this difference is statistically insignificant. Even though the preference inclines towards flicker, it is less noticeable compared to the Advanced group, and we consider this difference negligible. The other types of feedback (Average Continuous Rating and Overall rating from the end of questionnaire; not shown) confirm the trend of Comprehension for all groups.

4.4 Comprehension vs Continuous Rating

We collected Continuous Rating of the overall quality of subtitles at given times. For every comprehension question, we know the time span when the answer appears in the source speech document. Based on this timing information, we can relate comprehension and Continuous Rating. For a given time span answering a particular question, we find the most frequent Continuous Rating (button clicked most often) for every annotator. This gives us a histogram of Continuous Rating scores reported by different judges. In Figure 2 top, we show the correct ("OK") or partially correct answers ("OK-") and the histogram of Continuous Ratings by judges of distinct German proficiency levels. For a more detailed plot including all evaluation classes see Appendix B.1. This data aggregates observations for all documents and all setups excluding the offline SST and the oracle online SST without flicker.

For the judges with zero knowledge of German, we can not see any dependency of their comprehension to their Continuous Rating. On the other hand, the more the judges are proficient in German, the more their Continuous Rating reflects their comprehension. For example, for the C1 judges (Advanced) we can estimate their comprehension (and thus subtitle quality) from their clicking well: When they understand the content, the most probable given rating is 3 or 2. A less probable rating is 1, and they almost never rate 0 when they understand the content.

Listening while Rating In Figure 2 bottom, we show, from which source the judges knew the correct answer, either from the subtitles, or from sound. We can observe that indeed, the judges with German proficiency level B1 and higher listen to the source sound and understand, while the Zero level judges and Beginners rely only on subtitles.

Statistical Test To test the relation rigorously, we divide the judges into three groups by proficiency levels, their counts (see Table 2), their relation of Continuous Rating to correct answers and approach to listening versus reading (Figure 2). We run χ^2-test for statistical independence of Continuous Rating and answer results on the three groups. Test results are in Table 7. It shows that for the judges with Zero level of German, their Continuous Rating is independent on answer results. They do not follow the sound at all because they do not understand it, and rate only the readability and flicker. In case of the Beginners (A1 and A2, recall Table 2), we observe the dependency of their Continuous Rating on correct answers ("OK/OK-") and on cases when they did not answer ("unknown"). Their wrong answers and forgetting is independent of Continuous Rating, they probably make random mistakes uniformly. The Advanced group of judges give their correct, unknown or wrong answers consistently with their Continuous Rating. We therefore assume that they follow and understand the source speech and include the adequacy in their Continuous Rating.

We can also see that in all the three groups, the forgotten answers are independent on Continuous Rating. We assume that random and uniform out-

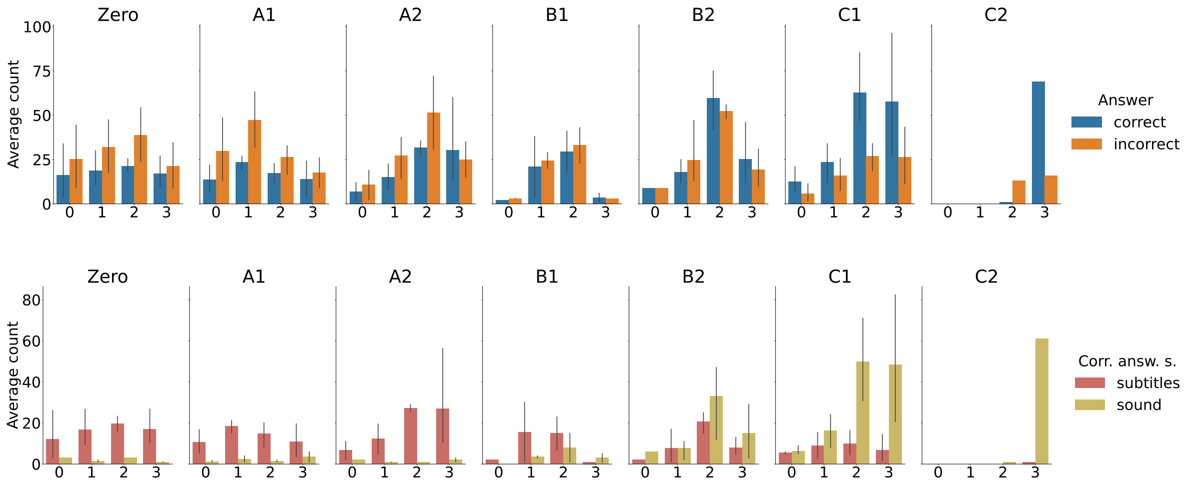


Figure 2: The average count of answers per judge for each proficiency level. Top: Correct (OK/OK-, blue bars) and incorrect (wrong/unknown, orange bars) answers vs Continuous Rating at the time when the answer was disclosed in the original document (x-axis, 0 means worst, 3 the best), distributed by source language proficiency level of the judges. Bottom: From which source the judges learned the correct or partially correct answer; subtitles in red, sound in yellow.

ages may be characteristic for human memory.

Practical Conclusions We conclude that Continuous Rating is a suitable for manual evaluation of simultaneous machine translation. The judges who speak the source language on at least B2 level on CEFR scale have an ability to assess SST quality reliably only by Continuous Rating, without the need for questionnaires which are laborious to prepare, answer and evaluate.

5 Conclusion

We proposed a novel and effective method for end-to-end user evaluation of simultaneous speech translation SST called Continuous Rating, publishing an open source evaluation tool for the future use. We showed that this method can be used for measuring comprehension and evaluating subtitling parameters. We demonstrated how user comprehension differs from offline MT to online MT. We showed that the users with a knowledge of the source language prefer low latency despite higher instability. We demonstrated that Continuous Rating can be used as a time-efficient human evaluation metric when employing judges with at least B2 (or, preferrably, C1) level of source language proficiency.

Limitations

This work is limited to only one direction of SST and lacks the comparison of multiple SST variants. Additionally, due to the number of investigated subtitling features and the smaller sample of judges, the results of layout experiments show only statistically insignificant preference towards one variant.

Acknowledgments

The research was partially supported by the grants 19-26934X (NEUREM3) of the Czech Science Foundation, “Grant Schemes at CU” (reg. no. CZ.02.2.69/0.0/0.0/19_073/0016935) and SVV project number 260 575.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, et al. 2021. Findings of the 2021 conference on machine translation (wmt21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. 2019. Monotonic infinite lookback attention for simultaneous machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1313–1323, Florence, Italy. Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, and George Foster. 2020. Re-translation versus streaming for simultaneous translation. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 220–227, Online. Association for Computational Linguistics.

Jan Berka, Ondrej Bojar, et al. 2011. Quiz-based evaluation of machine translation. *The Prague Bulletin of Mathematical Linguistics*, 95:77.

Colin Cherry and George Foster. 2019. Thinking slow about latency evaluation for simultaneous machine translation. *arXiv preprint arXiv:1906.00048*.

Eunah Cho, C. Fügen, T. Hermann, K. Kilgour, Mohammed Mediani, C. Mohr, J. Niehues, Kay Rottmann, C. Saam, Sebastian Stüker, and A. Waibel. 2013. A real-world system for simultaneous translation of german lectures. pages 3473–3477.

Eunah Cho, J. Niehues, and Alexander H. Waibel. 2012. Segmentation and punctuation prediction in speech language translation using a monolingual translation system. In *IWSLT*.

Florian Dessloch, Thanh-Le Ha, Markus Müller, Jan Niehues, Thai-Son Nguyen, Ngoc-Quan Pham, Elizabeth Salesky, Matthias Sperber, Sebastian Stüker, Thomas Zenkel, and Alexander Waibel. 2018. KIT lecture translator: Multilingual speech translation with one-shot learning. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 89–93, Santa Fe, New Mexico. Association for Computational Linguistics.

Maha Elbayad, Michael Ustaszewski, Emmanuelle Esperança-Rodier, Francis Brunet-Manquat, Jakob Verbeek, and Laurent Besacier. 2020. Online versus offline NMT quality: An in-depth analysis on English-German and German-English. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5047–5058, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Alvin Grissom II, He He, Jordan Boyd-Graber, John Morgan, and Hal Daumé III. 2014. Don't until the final verb wait: Reinforcement learning for simultaneous machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1342–1352, Doha, Qatar. Association for Computational Linguistics.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. 2017. Learning to translate in real-time with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1053–1062, Valencia, Spain. Association for Computational Linguistics.

Olivier Hamon, Christian Fügen, Djamel Mostefa, Victoria Arranz, Muntsin Kolss, Alex Waibel, and Khalid Choukri. 2009. End-to-end evaluation in simultaneous translation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 345–353, Athens, Greece. Association for Computational Linguistics.

Lifeng Han. 2018. Machine translation evaluation resources and methods: a survey. In *IPRC – Irish Postgraduate Research Conference*, Dublin, Ireland.

Javier Iranzo Sanchez, Jorge Civera, and Alfons Juan-Císcar. 2022. From simultaneous to streaming machine translation by leveraging streaming history. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6972–6985, Dublin, Ireland. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. 2019. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036, Florence, Italy. Association for Computational Linguistics.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, et al. 2018. Stacl: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. *arXiv preprint arXiv:1810.08398*.

Dominik Macháček and Ondřej Bojar. 2020. Presenting simultaneous translation in limited space. In *Proceedings of the 20th Conference Information Technologies – Applications and Theory (ITAT 2020), Hotel Tyrapol, Oravská Lesná, Slovakia, September 18-22, 2020*, volume 2718 of *CEUR Workshop Proceedings*, pages 34–39. CEUR-WS.org.

Markus Müller, Sarah Fünfer, Sebastian Stüker, and Alex Waibel. 2016a. Evaluation of the KIT lecture translation system. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1856–1861, Portorož, Slovenia. European Language Resources Association (ELRA).

Markus Müller, Thai Son Nguyen, Jan Niehues, Eunah Cho, Bastian Krüger, Thanh-Le Ha, Kevin Kilgour, Matthias Sperber, Mohammed Mediani, Sebastian Stüker, and Alex Waibel. 2016b. Lecture translator - speech translation framework for simultaneous lecture translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 82–86, San Diego, California. Association for Computational Linguistics.

Jan Niehues, Thai Son Nguyen, Eunah Cho, Thanh-Le Ha, Kevin Kilgour, Markus Müller, Matthias Sperber, Sebastian Stüker, and Alex Waibel. 2016. Dynamic transcription for low-latency speech translation. In *17th Annual Conference of the International Speech Communication Association, INTERSPEECH 2016*, volume 08-12-September-2016 of *Proceedings of the Annual Conference of the International Speech Communication Association. Ed. : N. Morgan*, pages 2513–2517. International Speech and Communication Association, Baixas.

Jan Niehues, Ngoc-Quan Pham, Thanh-Le Ha, Matthias Sperber, and Alex Waibel. 2018. Low-latency neural speech translation. In *Interspeech 2018*, Hyderabad, India.

Ofir Press and Noah A. Smith. 2018. You may not need attention. *CoRR*, abs/1810.13409.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

Hao Xiong, Ruiqing Zhang, Chuanqiang Zhang, Zhongjun Hea, Hua Wu, and Haifeng Wang. 2019. Dutongchuan: Context-aware translation model for simultaneous interpreting.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019. Simpler and faster learning of adaptive policies for simultaneous translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1349–1354, Hong Kong, China. Association for Computational Linguistics.

	Delay						
	70%	80%	90%	95%	99%	max	resets
ARS	**0.01**	**1.44**	**3.06**	**4.51**	**7.05**	**12.06**	8.80
FRS	1.74	3.54	5.18	7.52	10.65	16.78	**5.47**

Table 8: The adaptive reading speed (ARS) in comparison to the fixed reading speed (FRS), set to 18 char/sec. Percentages denote the proportion of words that have a delay less than the given number. The delay is in seconds, resets in the average count per document.

A Subtitler

A.1 Adaptive Reading Speed: Delay

We compared adaptive to fixed reading speed, averaging over all documents. We set the value of fixed reading speed to 18 characters per seconds, which we obtained by averaging all delays in the setting without adaptive reading speed.

The comparison is in Table 8. The delay was measured for all presented words. We used a subtitling window of 2 lines ×163 mm because it represents an upper bound for the delay of bigger subtitling windows.

B Results

B.1 Comprehensions vs Continuous Rating

In Figure 3, we show the average count of answers per judge for each proficiency level. Note two observations: 1) The number of already known answers is negligible, which proves that the questions were selected based on the content of documents. 2) The number of answers whose source was not given is high for all answers (Figure 3, right column), whereas it is low when correct and partially correct answers were selected (Figure 3, middle column). It means that judges provided the source when they answered a question.

B.2 Textual Feedback

In Table 9, we depict several textual ratings from Flicker Experiment. We select judges with C1 source language proficiency and contrast their feedback for flicker and no flicker.

The judges report higher satisfaction with flicker. They notice increased latency when the presentation mitigate flicker. This is consistent with our findings in Flicker experiment for Advanced group.

	Feedback
Setting	C1 proficiency, Overlay layout
Flicker	The subtitles weren’t so bad in terms of content or latency.
	The subtitles were very good, they just got stuck in the middle of the video, but after a short pause they worked again without any problems.
	The subtitles were relatively good, but despite their intelligibility and relative linguistic accuracy, they seemed very chaotic and very uncomfortable to read.
No flicker	A big delay of subtitles was sometimes inconvenient. If the subtitles are very delayed, it is almost impossible to follow them.
	The subtitles were small and dense, it was hard to orientate, especially when they were even delayed.
	At first, the delay was small. Then, at one point the subtitles got stuck and there was a lot of delay behind the sound.

Table 9: The selection of textual feedback from judges.

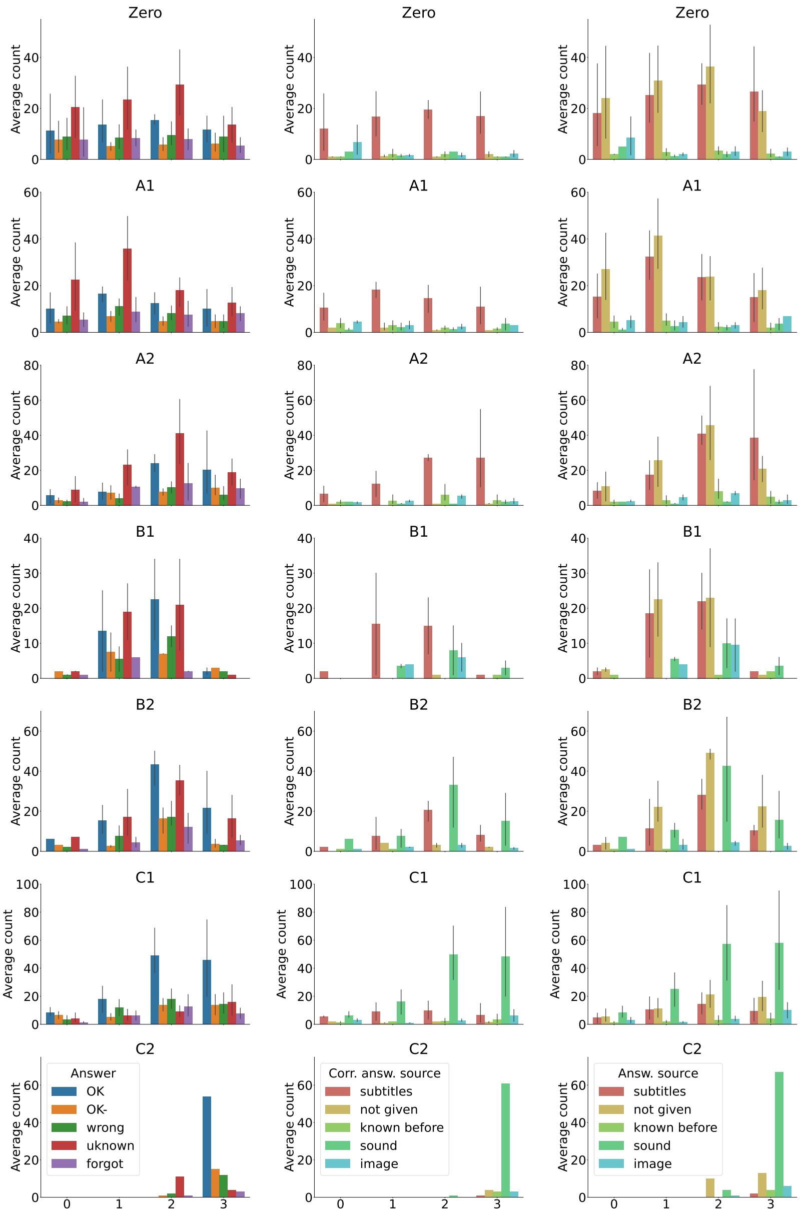


Figure 3: The average count of answers per judge for each proficiency level. Left: OK, OK-, wrong, unknown and forgotten answers vs Continuous Rating at the time when the answer was disclosed in the original document (x-axis, 0 means worst, 3 the best), distributed by source language proficiency level of the judges: from zero through beginners (A1, A2) and intermediate (B1, B2) to advanced (C1, C2). Middle: From which source the judges learned the correct (OK) or partially correct (OK-) answer. Right: From which source the judges learned all answers, regardless of their evaluation.

Gender Bias Mitigation for NMT Involving Genderless Languages

Ander Corral and **Xabier Saralegi**
Orai NLP Technologies, Basque Country, Spain
a.corral@orai.eus, x.saralegi@orai.eus

Abstract

It has been found that NMT systems have a strong preference towards social defaults and biases when translating certain occupations, which due to their widespread use, can unintentionally contribute to amplifying and perpetuating these patterns. In that sense, this work focuses on sentence-level gender agreement between gendered entities and occupations when translating from genderless languages to languages with grammatical gender. Specifically, we address the Basque to Spanish translation direction for which bias mitigation has not been addressed. Gender information in Basque is explicit in neither the grammar nor the morphology. It is only present in a limited number of gender specific common nouns and person proper names. We propose a template-based fine-tuning strategy with explicit gender tags to provide a stronger gender signal for the proper inflection of occupations. This strategy is compared against systems fine-tuned on real data extracted from Wikipedia biographies. We provide a detailed gender bias assessment analysis and perform a template ablation study to determine the optimal set of templates. We report a substantial gender bias mitigation (up to 50% on gender bias scores) while keeping the original translation quality.

1 Introduction

As the neural machine translation (NMT) field becomes more mature, there is a growing concern about the gender fairness of these systems (Stanovsky et al., 2019; Prates et al., 2020; Hovy et al., 2020; Savoldi et al., 2021). These data-driven approaches are trained on large real-world textual corpora which often exhibit implicit social gender stereotypes and biases. For example, Bolukbasi et al. (2016) noted that systems associate certain neutral occupations with males, such as doctor or programmer, and others with females, such as nurse or housekeeper. As a consequence, although not being required by the task, systems tend to inherit and amplify these social biases.

Several different solutions have been proposed to solve, or at least reduce, gender bias during the translation process: providing alternative masculine and feminine translations for some neutral words (Johnson, 2018); adding explicit gender information during training (Vanmassenhove et al., 2018; Stafanovičs et al., 2020; Saunders et al., 2020); removing bias from word embeddings (Font and Costa-Jussa, 2019); or fine-tuning on a small gender-balanced data set (Costa-jussà and de Jorge, 2020; Saunders et al., 2020). There have also been some efforts to construct some challenge sets to systematically assess gender bias (Stanovsky et al., 2019; Bentivogli et al., 2020).

Most of the previous work has focused on English as the source language which is then translated to languages with grammatical gender such as Spanish, French, German, etc. English is a notional gender language which encodes gender in a pronominal system (*he/she, his/her...*) (Savoldi et al., 2021). Consolidated evaluation benchmarks such as WinoMT (Stanovsky et al., 2019) or MuST-SHE (Bentivogli et al., 2020) are specially designed for English. However, for genderless languages such as Basque, existing previous work and benchmarks do not fully satisfy the requirements. For example, WinoMT uses pronominal references as a disambiguation signal for the correct inflection of occupations, which do not exist in Basque.

Gender information in Basque is explicit in neither the grammar nor the morphology. This fact implies that gender can only be determined when nouns correspond unequivocally to a female or a male, that is, person proper names or a limited number of gender-specific common nouns (e.g., *emakumea/gizona, aita/ama...*[1]), hereinafter referred to as gendered entities. Therefore, existing approaches and evaluation benchmarks need to be

[1] English translation: *woman/man, father/mother...*

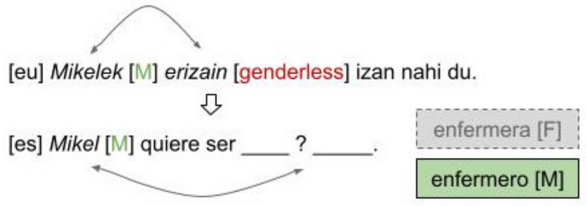

[eu] Mikelek [M] erizain [genderless] izan nahi du.
[es] Mikel [M] quiere ser ____ ? ____.
enfermera [F]
enfermero [M]

Figure 1: An illustrative example for the task of sentence-level agreement between gendered entities and occupations when translating from genderless languages to gendered languages. English translation: *Mikel wants to be a nurse.*

adapted to meet the requirements of a genderless language.

In this work we address the specific task of sentence-level gender agreement between gendered entities and occupations when translating from genderless languages to gendered languages (see example in Figure 1). We focus on the Basque to Spanish translation direction, a translation direction that presents the peculiarities described above and that has not been studied in the literature. The main contributions of the paper are the following:

- A template-based fine-tuning method with explicit gender signals to debias pre-trained systems involving genderless languages.
- A detailed experimentation to determine the source of gender bias for the task, including an in-depth ablation study of the template-based method and a comparison against fine-tuning on a gender-balanced Wikipedia biographies set.

2 Related work

A wide variety of publications warn about the lack of fairness some of the commercial MT systems have and how they might contribute to amplify and perpetuate social gender stereotypes due to their widespread use (Stanovsky et al., 2019; Prates et al., 2020; Hovy et al., 2020). In the case of the Basque-Spanish language pair, Salaberria et al. (2021) found a preference for the stereotyped translation of occupations according to their historically assigned role.

Ideally, system bias could be mitigated by removing all the bias present in the training data. For example, by augmenting data samples with their corresponding counterfactual forms (Zhao et al., 2018; Zmigrod et al., 2019). Nevertheless, this task still poses some challenges for grammatical gender languages as it involves preserving the morphosyntactic agreement of the whole sentence by gender swapping pronouns, adjectives, verbs, entities, etc., (Stafanovičs et al., 2020).

As a result, several alternative methods have been proposed to alleviate gender bias from MT systems. Vanmassenhove et al. (2018) add a special gender token to source sentences in order to improve morphological agreement between the uttered sentence and the gender of the speaker. Stafanovičs et al. (2020) annotate source words with the grammatical gender information of their corresponding target words. Basta et al. (2020) provide the system with discourse context by adding the previous sentence, and in the same direction, Moryossef et al. (2019) propose a method to guide a black-box model by appending some contextual gender unambiguous hints to source sentences, such as "... *she told them*". Other more complex approaches have targeted gender bias effects by directly equalizing genders in word embeddings (Escudé Font and Costa-jussà, 2019).

Another promising line of research addresses gender bias as a domain adaptation problem by fine-tuning a pre-trained biased system with a gender-balanced data set. Costa-jussà and de Jorge (2020) automatically collect gender-balanced parallel data from Wikipedia biographies by selecting an equal amount of examples for each gender. Choubey et al. (2021) generate gender filtered parallel data by forward-translating a monolingual corpus. Saunders and Byrne (2020) generate a small, trivial, gender-balanced set of synthetic examples by inflecting a single handcrafted template with an equal number of masculine and feminine entities. In Saunders et al. (2020) they further improve the method by adding explicit word-level gender tags.

Our proposed method uses a controlled gender-balanced set of examples (Section 3) to fine-tune a pre-trained model (Section 4.1). We further provide the system with explicit gender tags for proper gender inflections (Section 4.2). We assume that providing a stronger gender signal is better than letting the system infer the proper gender of each gendered entity. In order to annotate source words, Stafanovičs et al. (2020) propose a complex method involving morphological tagging and automatic alignment, and Saunders et al. (2020) rely on the proper coreference resolution. In contrast, we directly annotate gendered entities and leave gender agreement to the system instead of directly providing gender inflection information for all the affected words. This assumption simplifies the annotation effort as only gendered entities lists are necessary to annotate the data.

3 Gender-balanced corpora

Existing previous work on gender bias has focused on high resource translation directions (English to Spanish, French, German, etc.). While consolidated benchmarks and data exist for these languages, they strongly rely on gendered pronouns which makes them difficult to adapt for Basque. Thus, we analyzed two different strategies to build gender-balanced corpora for the Basque to Spanish translation direction: a syntactically diverse set of handcrafted templates (Section 3.1), and real data extracted from Wikipedia biographies (Section 3.2).

3.1 Handcrafted templates

A Basque and Spanish native speaker manually designed a set of task-specific templates for the correct treatment of gender agreement at sentence-level between gendered entities and occupations. We argue that a single tiny template does not provide sufficient syntactic diversity, so we handcrafted a syntactically diverse set of 33 templates. Each of the templates has placeholders for an occupation and a gendered entity to help in the proper disambiguation of that occupation.

Saunders et al. (2020) reported that systems trained on single-entity templates tend to overgeneralize gender signals on multi-entity examples by indiscriminately applying the same gender to all the occupations regardless of the other entities' genders. For instance, the Basque source sentence "*Josean iragarlea zen eta Leirek idazkaria izan nahi zuen.*" [2] would be translated to "*Josean era adivino y Leire quería ser secretario.*" instead of producing the correct feminine form "*secretaria*". To address this issue, we also construct a set of 13 multi-entity templates with two gendered entities and their corresponding occupations.

We use an occupations list and a gendered entities list to automatically populate the templates. We slightly adapted the list of occupations from Salaberria et al. (2021) to obtain a set of 83 occupations in Basque and their respective translations for both genders in Spanish. The gendered entities list contains a set of 200 common Basque and Spanish person proper names. We further complemented that list with 14 gendered common nouns referring to humans in Basque (e.g., *emakumea/gizona, aita/ama...*[3]) by querying Basque WordNet (Pociello et al., 2011). We collected the same amount of gendered entities for each gender.

We randomly divided the handcrafted templates[4] into disjoint training and test sets, keeping 6 single entity templates and 3 multi-entity templates for testing purposes. For each gender, 20 proper names and 4 gendered terms are used to inject these test templates. A total amount of 3,120 single entity examples and 1,800 multi-entity examples were created, hereinafter referred to as *Templ_test* and *Multi_test* test sets. The rest, 27 and 10 templates respectively, are kept to create training data (*Templ_train* and *Multi_train*) and were injected in different ways as explained in Section 4.1. Some examples of the handcrafted templates are shown in Table 1.

3.2 Back-translated Wikipedia biographies

In order to generate a gender-balanced set of real data, we turned to Wikipedia biographies. We focused on the extraction of examples that present gender agreement between people and occupations for the Basque-Spanish language pair. Unlike the strategy proposed by Costa-jussà et al. (2019), we extract task specific examples from Spanish monolingual biographies which are then back-translated to Basque, instead of directly extracting parallel data. The reason behind this decision was that more task specific examples could be gathered from leveraging Spanish monolingual data only, as Basque biographies constrained the amount of examples that could be gathered.

We searched the Spanish Wikipedia (extracted using WikiExtractor[5]) for biographies of living persons using Petscan[6]. We found 160,641 biographies matching this criteria. Using the Wikidata API, we automatically detected the gender of these persons. We only extracted the first sentence from each biography, which generally includes examples

[2] English translation: *Josean was a fortune-teller and Leire wanted to be a secretary.*

[3] English translation: *woman/man, father/mother...*

[4] All the handcrafted templates and occupations and gendered entities lists are included in the supplementary material.

[5] https://github.com/attardi/wikiextractor

[6] https://petscan.wmflabs.org/

SINGLE-ENTITY TEMPLATE
eu: {entity}k {occupation} izan nahi du.
→ *Mikelek erizain izan nahi du.*
es: {entity} quiere ser {occupation}.
→ *Mikel quiere ser enfermero*
en: {entity} wants to be a {occupation}.
→ *Mikel wants to be a nurse.*

SINGLE-ENTITY TEMPLATE
eu: Nire lagun {entity} {occupation}a zela esan nizunean haserratu egin zinen.
→ *Nire lagun Ainara errementaria zela esan nizunean haserratu egin zinen.*
es: Cuando te dije que mi amigo|a {entity} era {occupation} te enfadaste.
→ *Cuando te dije que mi amiga Ainara era herrera te enfadaste.*
en: When I told you my friend {entity} was a {occupation} you got angry.
→ *When I told you my friend Ainara was a blacksmith you got angry.*

MULTI-ENTITY TEMPLATE
eu: {entity}k {occupation} izatea gustuko du, baina {entity2}k {occupation2} izatea gorroto du.
→ *Mikelek erizain izatea gustuko du, baina Ainarak errementari izatea gorroto du.*
es: A {entity} le gusta ser {occupation}, pero {entity2} odia ser {occupation2}.
→ *A Mikel le gusta ser enfermero, pero Ainara odia ser herrera.*
en: {entity} loves being a {occupation} while {entity2} hates being a {occupation2}.
→ *Mikel loves being a nurse while Ainara hates being a blacksmith.*

Table 1: Examples of the handcrafted templates for the gender agreement task between gendered entities and occupations. Along with the templates we provide an injected example and the corresponding English translation.

of gender agreement between persons and occupations. For example:

***Elisabeth Rynell** es una **escritora** sueca que ha incursionado principalmente en los géneros de la novela y poesía.*[7]

Finally, the extracted examples were automatically back-translated to Basque with the baseline system described in Section 4. This process guarantees gender agreement is not altered during the translation process as Basque is a genderless language.

We obtained a final set of approximately 42,000 examples per gender with the required gender agreement (*Wiki_train*).

In addition, in order to have an in-domain test from Wikipedia, we manually created a disjoint test set by selecting 100 examples for each gender. In this case, translations were manually corrected to ensure their final quality. Hereinafter referred to as *Wiki_test* test set.

4 Experimentation

All the systems use the default configuration for the Transformer architecture (Vaswani et al., 2017) as implemented in the PyTorch version of the OpenNMT toolkit (Klein et al., 2017). We apply BPE tokenization (Sennrich et al., 2016) trained on 32,000 operations on the joint training data. Sentences larger than 100 tokens are discarded from the training set.

The baseline systems were trained on the Basque-Spanish portion (1.77M examples) of the Paracrawl (v8) data (Bañón et al., 2020). The gender-balanced systems are trained by fine-tuning the baseline system on the gender-balanced data sets described in Section 3. As in Costa-jussà and de Jorge (2020), to avoid catastrophic forgetting, where systems tend to forget about previous knowledge, we follow a mixed fine-tuning strategy (Chu et al., 2017). A weighted combination (10:1 ratio[8]) of general domain data from Paracrawl and task specific data, such as *Templ_train* or *Wiki_train*, is used during training and validation steps. For

[7] English translation: *Elisabeth Rynell is a Swedish writer who has mainly dabbled in the genres of novels and poetry.*

[8] Initial experiments showed that 10:1 ratio for general domain and task specific data respectively works well.

validation purposes, we concatenate 5,000 general domain examples and 1,000 task specific examples randomly extracted from the training data.

The baseline and the fine-tuned systems have been trained until convergence on the perplexity results on the validation set, stopping the training process if there was no improvement for 5 consecutive checkpoints. Validation is performed every 10,000 steps in the case of the baseline system whereas fine-tuning validation is performed every 1,000 steps.

We evaluate our systems using **BLEU** and **chrF++** scores from the sacreBLEU tool (Post, 2018). Additionally, we also provide **COMET** (Rei et al., 2020) scores[9], a metric which focuses on the semantic similarity by leveraging the recent breakthroughs in neural language modeling. These scores are computed on the test sets extracted from three publicly available corpora: **EiTB** (Etchegoyhen and Gete, 2020) a news domain data set, **EhuHac** (Sarasola et al., 2015) a collection of classic books, and **TED** (Reimers and Gurevych, 2020) comprising TED talks transcriptions. From each set, we randomly extracted 5,000 examples. Although BLEU, chrF++ and COMET metrics measure the overall translation quality of the systems, task specific metrics are required to evaluate gender bias with more precision. To that end, we measure the **accuracy** of the correctly translated and gender inflected occupations and we propose a new metric called **swap**. Swap is defined as the percentage of the occupations which are inflected with the opposite gender. Thus, errors are divided into unrecoverable errors where occupations are translated in a different way (errors) and gender swapped occupations (swap). A higher swap score means higher bias towards the opposite gender. These scores are computed on the task specific test sets mentioned in Section 3.

4.1 Gender bias assessment

We conducted a detailed experimentation to determine the source of gender bias in the agreement between gendered entities and occupations. We analyze four different strategies to inject different subsets of the gendered entities and occupations lists in the training templates in order to generate gender-balanced data:

- **Full** system uses all the available gendered entities and occupations, both training and test subsets, to inflect training templates. Therefore, the test subsets of the gendered entities and occupations are seen during training. We produce 772,896 training examples.

- **Unknown entities (Unk_ent)** system only uses the training subset of the entities to inflect training templates. There is no overlap between the entities used for fine-tuning the system and those for the test set. 693,880 training examples are produced.

- **Unknown occupations (Unk_occ)** system only uses the training subset of the occupations to inflect training templates, resulting in 633,216 training examples.

- **Unknown pairs (Unk_prs)** system only inflects templates with disjoint combinations of entities and occupations. For instance, if *Mikel-doctor* is present in the test, *Mikel-plumber* and *Jon-doctor* are seen during training. A total of 758,616 training examples are produced.

We remark that, in all the cases, training (27) and test (6) template sets are disjoint, and only single entity templates are used for fine-tuning.

In general terms, all the fine-tuned systems on gender-balanced data keep the baseline's translation quality on the general domains test sets (see Table 2). Specially, **Full** system performs at par with the baseline across all the test sets and metrics, except for the chrF++ score on the EhuHac test set.

In order to analyze bias effects, in Table 3 we report gender bias accuracy and swap scores. The baseline model shows a clear bias towards the masculine inflection of the occupations with significantly higher swap scores and lower accuracy scores for females. We note that a negative value for the swap difference means there exists masculine bias. Lower swap scores are obtained with the baseline system on the *Wiki_test*, suggesting stereotyped occupations from Wikipedia (Wagner et al., 2015) are being well inflected by the baseline.

Fine-tuning the baseline system on the **Full** set significantly drops the swap score on the *Templ_test* which indicates the fine-tuned system is able to correctly inflect the gender of the occupations for seen entities. In contrast, **Unk_ent** system shows higher swap scores. Despite the system is clearly less biased than the baseline, it is having

[9] The recommended model *wmt20-comet-da* was used and it already covers both Basque and Spanish.

System	EiTB			EhuHac			TED		
	BLEU	chrF++	COMET	BLEU	chrF++	COMET	BLEU	chrF++	COMET
Baseline	**37.9**	**57.8**	**0.732**	**14.1**	**37.0**	**-0.149**	**24.2**	**48.6**	**0.462**
Full	**37.8**	**57.9**	**0.730**	**14.0**	36.8*	**-0.153**	**24.1**	**48.6**	**0.458**
Unk_ent	**37.8**	57.7*	0.725*	**14.0**	36.9*	**-0.154**	**24.1**	**48.6**	0.456*
Unk_occ	37.8*	**57.8**	0.725*	**14.0**	36.8*	**-0.152**	24.0*	48.5*	0.456*
Unk_prs	37.7*	57.7*	**0.727**	**14.0**	**36.9**	**-0.152**	**24.0**	**48.6**	0.456*

Table 2: BLEU, chrF++ and COMET scores for systems fine-tuned on gender-balanced data. * indicates statistically significant (p-value $\leq$ 0.05) differences by conducting paired bootstrap resampling with respect to the baseline. Best scoring systems are highlighted in bold.

System	Test	Male		Female		$\bar{X}$ Swap	Δ Swap
		Acc	Swap	Acc	Swap		
Baseline	Templ_test	59.23	0.77	14.04	47.31	24.04	-46.54
	Multi_test	61.78	**6.17**	17.72	50.67	28.42	-44.50
	Wiki_test	**95.15**	**0.61**	65.82	25.95	13.00	-25.34
Full	Templ_test	**98.27**	0.96	**96.15**	**2.82**	**1.44**	**-1.86**
	Multi_test	76.22	23.72	**83.39**	16.61	20.17	7.11
	Wiki_test	93.94	1.21	67.09	24.68	12.69	-23.47
Unk_ent	Templ_test	93.72	5.51	83.33	15.71	10.61	-10.2
	Multi_test	67.39	32.56	76.83	23.11	27.83	9.45
	Wiki_test	93.33	**0.61**	67.09	**24.05**	**12.07**	-23.44
Unk_occ	Templ_test	62.95	**0.45**	58.91	6.41	3.43	-5.96
	Multi_test	56.28	12.56	55.61	**15.06**	**13.81**	**-2.50**
	Wiki_test	93.33	1.21	67.09	24.68	12.69	-23.47
Unk_prs	Templ_test	97.95	0.96	94.10	5.00	2.98	-4.04
	Multi_test	**76.56**	23.22	81.00	18.94	21.08	4.28
	Wiki_test	92.73	1.21	**67.72**	**24.05**	12.38	**-22.84**

Table 3: Accuracy and swap scores for systems fine-tuned on gender-balanced data. We report mean swap scores and swap differences for a better picture of the bias. Best scoring systems are highlighted in bold.

more difficulties inferring the gender of unseen entities. This behaviour is further corroborated on the *Wiki_test* as all the experiments show similar (slightly better) results of those obtained by the baseline system. A manual inspection of the translations showed that most of the swap errors are associated to foreign person names. In such cases, systems tend to provide the default masculine.

With respect to the lower accuracy scores in **Unk_occ**, most of the errors made were the result of the system not being able to produce the correct translation. Most of the time it produces correct but alternative translations for the given occupations such as *lechero/vendedor de leche (milkman)*. In any case, we remark that in these cases the correct gender is generally inflected: *lechera/vendedora de leche (milkwoman)* or *camarógrafo/el cámara (cameraman)*. This behavior blurs swap and accuracy results as the bias can not be automatically computed for those occupations.

Remarkably, accuracy and swap scores in **Unk_prs** obtains comparable enough results of those obtained in **Full**, which suggests the system does not require to see all the possible combinations of entities and occupations during training. Instead, in the light of the results obtained by the **Unk_ent** and **Unk_occ** systems, providing the system with all the gendered entities and occupations is more relevant than producing all their combinations.

In general terms, we can conclude that fine-tuning a pre-trained system with a mixed combination of gender-balanced examples and general domain data is useful to mitigate gender bias from NMT systems without a substantial drop in general domain translation quality.

Finally, we note that all the experiments show higher swap scores on the *Multi_test* test. A manual inspection of the translations suggests that, as stated in (Saunders et al., 2020), systems simply learn to indiscriminately apply the same gender inflection to all the occupations when presented with multi-entity templates. This issue is addressed in Section 4.2.

4.2 Gender tagging entities

From the previous bias assessment section, we conclude that gender disambiguation for unknown entities is not obvious for the systems. Yet, it is essential to correctly inflect occupations with the corresponding gender. Therefore, in the line of the previous work by Saunders et al. (2020) and Stafanovičs et al. (2020), we propose using word level annotations to provide a stronger gender signal for gender disambiguation of gendered entities. (Stafanovičs et al., 2020) annotate all the source words with the grammatical gender information of their corresponding target words while (Saunders et al., 2020) add explicit word-level gender tags to the occupations that need to be inflected. In contrast to these methods, we only apply gender annotations to gendered entities and leave sentence-level gender agreement to the system. The main advantage of this approach is that it does not require any complex annotation step. During inference, a list of proper names and other gendered entities can be used to properly annotate the entities. This list can be dynamically updated without fine-tuning the whole system again.

We annotate each word in the source side of the **Full** set via source factors (Sennrich and Haddow, 2016) with three possible values (**Full_tag**): 1 for male entities, 2 for female entities and 0 for the rest of the words. For example,

Mikelek erizain izan nahi du.[10] → 1 0 0 0 0

These tags are then appropriately mapped to their corresponding subword tokens during the fine-tuning step.

Additionally, as noted in Saunders et al. (2020) and corroborated in Section 4.1, in cases where multiple entities are present in a sentence, systems tend to overgeneralize gender signals by applying the same gender to all the occupations. Accordingly, we add the *Multi_train* set (see Section 3.1) during the fine-tuning step to help the **Full_tag** system better handle these cases.

Likewise, we follow the same gender tagging strategy on the Wikipedia biographies set (*Wiki_train*), described in Section 3.2, to assess whether using real data extracted from Wikipedia is a feasible approach. We remark that due to the characteristics of the biographies it is not possible to extract multiple entity examples. We fine-tune the baseline system with (**Wiki_tag**) and without gender tags (**Wiki**).

Overall, all the systems keep the baseline's translation quality in terms of BLEU, chrF++ and COMET for the general domain test sets, either fine-tuned with templates or with real Wikipedia data (see Table 4).

Moreover, Table 5 shows gender bias accuracy and swap scores for the gender tagged systems along with their untagged version. **Full_tag** considerably outperforms **Full**. Despite the swap difference on the *Templ_test* is slightly higher for **Full_tag**, the total swap score is lower. Adding unambiguous gender tags to the entities provides a stronger signal that helps reducing gender bias from the system. We report a substantial improvement on the *Multi_test*, which further encourages the use of a stronger gender signal via gender tags. Remarkably, **Full_tag** obtains perfect scores on the *Multi_test*, showing that providing multi-entity templates during training helps mitigating the gender signal overgeneralization issue.

Fine-tuning on *Wiki_train* also helps improving the baseline system and adding gender tags further improves those results. As expected, **Wiki_tag** obtains the best bias reduction results on the in-domain *Wiki_test*, as most of the occupations overlap between training and test data. Notice that the Wikipedia occupations set is potentially small and closed. As reported by Costa-jussà and de Jorge (2020) using real gender-balanced data instead of manually created templates can also contribute to reduce bias, although the results we achieved are not as good as those obtained with the template-based version. However, we note that systems fine-tuned on *Wiki_train* still perform poorly when multiple entities are present, showing a clear tendency towards masculine overgeneralization.

4.3 Templates ablation study

In this section we report the results of the template ablation experiments conducted to determine the optimal amount of templates needed in order to reduce the manual effort to build them. We focused on their complexity too, as generating simpler templates might be easier without a strong knowledge

[10] English translation: Mikel wants to be a nurse.

System	**EiTB**			**EhuHac**			**TED**		
	BLEU	chrF++	COMET	BLEU	chrF++	COMET	BLEU	chrF++	COMET
Baseline	**37.9**	**57.8**	**0.732**	14.1	**37.0**	**-0.149**	**24.2**	**48.6**	**0.462**
Full	**37.8**	**57.9**	**0.730**	14.0	36.9*	**-0.153**	**24.1**	**48.6**	**0.458**
Full_tag	**37.8**	**57.8**	**0.729**	14.0	36.9*	**-0.150**	**24.1**	**48.6**	**0.457**
Wiki	**38.0**	**57.9**	**0.733**	**14.2***	**37.0**	**-0.143**	**24.1**	**48.6**	**0.461**
Wiki_tag	**37.9**	**57.8**	**0.731**	14.1	**36.9**	**-0.147**	**24.1**	**48.6**	**0.461**

Table 4: BLEU, chrF++ and COMET scores for the gender tagging systems compared to their untagged versions. * indicates statistically significant (p-value $\leq$ 0.05) differences by conducting paired bootstrap resampling with respect to the baseline. Best scoring systems are highlighted in bold.

System	**Test**	**Male**		**Female**		**$\bar{X}$ Swap**	**Δ Swap**
		Acc	**Swap**	**Acc**	**Swap**		
Baseline	Templ_test	59.23	0.77	14.04	47.31	24.04	-46.54
	Multi_test	61.78	6.17	17.72	50.67	28.42	-44.5
	Wiki_test	95.15	0.61	65.82	25.95	13.00	-25.34
Full	Templ_test	98.27	0.96	96.15	2.82	1.44	**-1.86**
	Multi_test	76.22	23.72	83.39	16.61	20.17	7.11
	Wiki_test	93.94	1.21	67.09	24.68	12.69	-23.47
Full_tag	Templ_test	**99.49**	**0.06**	**96.86**	**2.12**	**1.09**	-2.06
	Multi_test	**100.00**	**0.00**	**100.00**	**0.00**	**0.00**	**0.00**
	Wiki_test	95.15	**0.00**	84.81	5.70	2.79	-5.70
Wiki	Templ_test	57.37	2.24	20.38	40.13	21.19	-37.89
	Multi_test	60.11	7.00	19.78	49.22	28.11	-42.22
	Wiki_test	**95.76**	**0.00**	76.58	15.82	7.74	-15.82
Wiki_tag	Templ_test	62.95	0.32	37.50	25.32	12.82	-25.00
	Multi_test	60.39	4.56	17.67	50.00	27.28	-45.44
	Wiki_test	**95.76**	**0.00**	**92.41**	**1.90**	**0.93**	**-1.90**

Table 5: Accuracy and swap scores for the gender tagged systems compared to their untagged versions on the task specific test sets. We report mean swap scores and swap differences for a better picture of the bias. Best scoring systems are highlighted in bold.

about the language.

We sorted all the training templates, both single entity templates and multi-entity templates, according to their complexity. Word counts are used as an indicator of their complexity. We wanted to analyze the simplest scenario with just one single entity template and one multi-entity template (**1_1**), as this is the case in (Saunders and Byrne, 2020). Additionally, we analyzed scenarios with different numbers of single entity templates and multi-entity templates, hereinafter referred to as **2_2**, **5_5**, **10_10**, **20_10**[11]. To analyze the influence of the complexity, for all the combinations we produced a simple version (**S**), which comprises the less complex templates and a complex version (**C**) including the most complex ones. All these ablation experiments were compared against the baseline system and the **Full_tag** system fine-tuned with all the handcrafted templates possible (27 single entity and 10 multi-entity templates). All the experiments use gender tags.

In general terms, all the systems comply with the requirement of keeping the baseline's translation quality for the general domain test sets (see Table 6). We therefore focus on the task specific metrics as shown in Figure 2.

All the systems, even for **S_1_1**, significantly improve swap scores when compared to the baseline. Mean swap curves show a clear descending trend which suggests that having a more syntactically diverse set helps generalizing gender signals. Remarkably, from **S_10_10** and **C_10_10** systems on, the curve tends to converge, showing little improvement with more templates. This is an interesting

[11] Names indicate the number of single entity and multi-entity templates respectively

System	EiTB			EhuHac			TED		
	BLEU	chrF++	COMET	BLEU	chrF++	COMET	BLEU	chrF++	COMET
Baseline	**37.7**	**57.8**	**0.732**	**14.1**	**37.0**	**-0.149**	**24.2**	**48.6**	**0.462**
S_1_1	**37.8**	**57.8**	**0.728**	**14.0**	36.9*	**-0.148**	23.9*	48.5*	**0.458**
C_1_1	37.7*	**57.7**	**0.728**	13.9*	36.8*	**-0.154**	**24.1**	**48.7**	**0.462**
S_2_2	**37.8**	**57.8**	**0.727**	**14.0**	**36.9**	**-0.152**	**24.0**	48.5*	**0.459**
C_2_2	37.7*	57.7*	**0.727**	**14.0**	36.8*	**-0.155**	23.9*	48.5*	0.456*
S_5_5	37.7*	57.7*	0.725*	**14.0**	**36.9**	**-0.152**	24.0*	48.5*	0.456*
C_5_5	**37.8**	**57.8**	**0.731**	**14.0**	36.8*	**-0.149**	23.9*	**48.4**	0.454*
S_10_10	**37.8**	**57.8**	**0.727**	**14.0**	**36.9**	**-0.153**	**24.2**	**48.7**	**0.461**
C_10_10	**37.8**	**57.8**	**0.729**	**14.0**	36.8*	**-0.152**	**24.0**	48.5*	0.456*
S_20_10	**37.8**	**57.7**	**0.727**	**14.0**	36.8*	**-0.153**	24.0*	**48.6**	**0.461**
C_20_10	**37.8**	57.7*	**0.726**	**14.0**	36.8*	**-0.150**	23.9*	48.5*	0.455*
Full_tag	**37.8**	**57.8**	**0.729**	**14.0**	36.9*	**-0.150**	**24.1**	**48.6**	**0.457**

Table 6: BLEU, chrF++ and COMET scores for the template ablation experiments. * indicates statistically significant (p-value $\leq$ 0.05) differences by conducting paired bootstrap resampling with respect to the baseline. Best scoring systems are highlighted in bold.

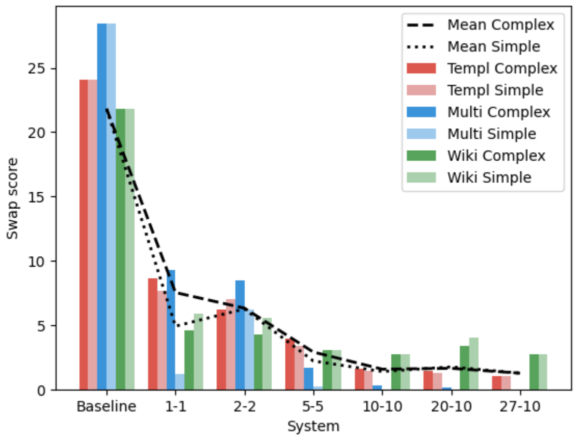


Figure 2: Total swap scores for the template ablation experiments. Dashed and doted lines represent the mean swap values for the three test sets (*Templ_test*, *Multi_test* and *Multi_test*).

insight, as there seems to be a limit in the amount of templates, which considerably reduces the manual effort to create templates.

In general terms, systems fine-tuned on simpler templates perform at par or even better than the ones trained on more complex templates. This indicates that generating complex and syntactically rich templates is not worth the effort. Also, results suggest that multi-entity templates present a strong signal which solves the overgeneralization issue for systems with 10 or more templates. Thus, it emphasizes the hypothesis that templates can be easily adapted to small specific tasks with little effort.

System	Templ	Multi	Multi
S_1_1	7.72	1.22	5.88
S_10_10	**1.44**	**0.03**	**2.79**
S_10_10_limit	1.70	0.25	3.41

Table 7: Mean swap scores for the **S_10_10_limit** system compared against its complete version (**S_10_10**) and **S_1_1**. The system was fine-tuned on only 32,200 examples as **S_1_1**. Best scoring systems are highlighted in bold.

Finally, we tested whether having more templates improves the results because of the syntactically diverse templates or just because of the mere fact that more training examples are generated. To that end, as we observed some convergence with 10 templates, we fine-tuned the baseline on randomly selected 32,200 examples, namely **S_10_10_limit**, that is, the same amount of templates used in **S_1_1** (Table 7). **S_10_10_limit** performs slightly worse than **S_10_10** and it clearly outperforms **S_1_1**. This suggests that the improvement comes to a greater extent from syntactic diversity rather than from a higher amount of templates.

5 Conclusions

In this work we addressed gender bias mitigation from an already pre-trained system. In particular, we focused on the specific task of sentence-level gender agreement between gendered entities and occupations when translating from genderless languages to gendered languages.

The proposed template-based fine-tuning strategy with explicit gender tags helps mitigating gender bias from NMT systems. We proved that the mixed fine-tuning strategy using a weighted combination of general domain and task specific data is beneficial to overcome catastrophic forgetting and keep the original translation quality.

We demonstrated that adding explicit gender tags to gendered entities provides a stronger gender signal and helps the system to gender inflect occupations correctly. At inference, entities can be easily annotated by using a list of proper names and other gendered entities, which can be dynamically updated without fine-tuning the system again.

Our results on the Basque to Spanish translation direction showed substantial bias mitigation and confirmed that handcrafted templates are suitable to create task specific training examples, to the point of improving the results obtained by using gender-balanced real examples extracted from Wikipedia. The ablation study showed that with little manual effort a set of useful templates can be created for gender bias mitigation. Therefore, the proposed method can be applied to other language pairs.

Limitations

The ablation study in Section 4.3 showed that with little manual effort a set of useful templates could be created for gender bias mitigation. In this sense, the proposed method still requires some linguistic knowledge about the languages involved in order to manually create the templates. Some of the entities and the occupations list should be adapted to the new language pair too. We acknowledge that this requirement can be a limiting factor for a massive deployment of our method. However, we believe that some challenges in NMT require a prior linguistic knowledge of the languages at hand in order to detect the possible errors and flaws and to provide a solution or mitigation response.

Furthermore, our work focuses on the Basque to Spanish translation direction as an example of the translation direction from a genderless language to a language with explicit grammatical gender. Although, the proposed gender tagging method does not rely on Basque or Spanish exclusive linguistic features, we believe that adding additional language pairs would have shown a broader picture of our method's potential. We leave this discussion for future work where different language families could be analyzed.

Finally, it must be noted that we approach the task using a binary gender representation schema. This decision should not be interpreted as a denial of a more complex reality.

Acknowledgments

This work has been partially funded by the Basque Government (TANPER, Hazitek grant no. ZL-2021/00904)

References

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4555–4567, Online. Association for Computational Linguistics.

Christine Basta, Marta R. Costa-jussà, and José A. R. Fonollosa. 2020. Towards mitigating gender bias in a decoder-based neural machine translation model by adding contextual information. In Proceedings of the The Fourth Widening Natural Language Processing Workshop, pages 99–102, Seattle, USA. Association for Computational Linguistics.

Luisa Bentivogli, Beatrice Savoldi, Matteo Negri, Mattia A. Di Gangi, Roldano Cattoni, and Marco Turchi. 2020. Gender in danger? evaluating speech translation technology on the MuST-SHE corpus. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6923–6933, Online. Association for Computational Linguistics.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. Advances in neural information processing systems, 29.

Prafulla Kumar Choubey, Anna Currey, Prashant Mathur, and Georgiana Dinu. 2021. GFST: Gender-filtered self-training for more accurate gender in translation. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 1640–1654, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short

Papers), pages 385–391, Vancouver, Canada. Association for Computational Linguistics.

Marta R. Costa-jussà and Adrià de Jorge. 2020. Fine-tuning neural machine translation on gender-balanced datasets. In Proceedings of the Second Workshop on Gender Bias in Natural Language Processing, pages 26–34, Barcelona, Spain (Online). Association for Computational Linguistics.

Marta R Costa-jussà, Pau Li Lin, and Cristina España-Bonet. 2019. Gebiotoolkit: Automatic extraction of gender-balanced multilingual corpus of wikipedia biographies. arXiv preprint arXiv:1912.04778.

Joel Escudé Font and Marta R. Costa-jussà. 2019. Equalizing gender bias in neural machine translation with word embeddings techniques. In Proceedings of the First Workshop on Gender Bias in Natural Language Processing, pages 147–154, Florence, Italy. Association for Computational Linguistics.

Thierry Etchegoyhen and Harritxu Gete. 2020. Handle with care: A case study in comparable corpora exploitation for neural machine translation. In Proceedings of the 12th Language Resources and Evaluation Conference, pages 3792–3800. European Language Resources Association.

Joel Escudé Font and Marta R Costa-Jussa. 2019. Equalizing gender biases in neural machine translation with word embeddings techniques. arXiv preprint arXiv:1901.03116.

Dirk Hovy, Federico Bianchi, and Tommaso Fornaciari. 2020. "you sound just like your father" commercial machine translation systems include stylistic biases. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1686–1690.

Melvin Johnson. 2018. Providing gender-specific translations in google translate. Accessed: 2022-02-17.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In Proceedings of ACL 2017, System Demonstrations, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Amit Moryossef, Roee Aharoni, and Yoav Goldberg. 2019. Filling gender & number gaps in neural machine translation with black-box context injection. In Proceedings of the First Workshop on Gender Bias in Natural Language Processing, pages 49–54, Florence, Italy. Association for Computational Linguistics.

Elisabete Pociello, Eneko Agirre, and Izaskun Aldezabal. 2011. Methodology and construction of the basque wordnet. Language resources and evaluation, 45(2):121–142.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In Proceedings of the Third Conference on Machine Translation: Research Papers, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Marcelo OR Prates, Pedro H Avelar, and Luís C Lamb. 2020. Assessing gender bias in machine translation: a case study with google translate. Neural Computing and Applications, 32(10):6363–6381.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 2685–2702, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics.

Ander Salaberria, Jon Ander Campos, Iker García, and Joseba Fernandez de Landa. 2021. Itzulpen automatikoko sistemen analisia: Genero alborapenaren kasua. In Fourth Conference for Basque Researchers.

Ibon Sarasola, Pello Salaburu, and Josu Landa. 2015. Hizkuntzen arteko corpusa (hac).

Danielle Saunders and Bill Byrne. 2020. Reducing gender bias in neural machine translation as a domain adaptation problem. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7724–7736, Online. Association for Computational Linguistics.

Danielle Saunders, Rosie Sallis, and Bill Byrne. 2020. Neural machine translation doesn't translate gender coreference right unless you make it. In Proceedings of the Second Workshop on Gender Bias in Natural Language Processing, pages 35–43, Barcelona, Spain (Online). Association for Computational Linguistics.

Beatrice Savoldi, Marco Gaido, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2021. Gender bias in machine translation. Transactions of the Association for Computational Linguistics, 9:845–874.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers, pages 83–91, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In Proceedings of the 54th Annual Meeting of the Association for Computational

Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Artūrs Stafanovičs, Toms Bergmanis, and Mārcis Pinnis. 2020. Mitigating gender bias in machine translation with target gender annotations. In Proceedings of the Fifth Conference on Machine Translation, pages 629–638, Online. Association for Computational Linguistics.

Gabriel Stanovsky, Noah A Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1679–1684.

Eva Vanmassenhove, Christian Hardmeier, and Andy Way. 2018. Getting gender right in neural machine translation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3003–3008, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. Advances in neural information processing systems, 30.

Claudia Wagner, David Garcia, Mohsen Jadidi, and Markus Strohmaier. 2015. It's a man's wikipedia? assessing gender inequality in an online encyclopedia. In Proceedings of the international AAAI conference on web and social media, volume 9, pages 454–463.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1651–1661, Florence, Italy. Association for Computational Linguistics.

Exploring the Benefits and Limitations of Multilinguality for Non-autoregressive Machine Translation

Sweta Agrawal[1] **and Julia Kreutzer**[2] **and Colin Cherry**[2]
[1]Department of Computer Science, University of Maryland
[2]Google Research
sweagraw@umd.edu, {jkreutzer, colincherry}@google.com

Abstract

Non-autoregressive (NAR) machine translation has recently received significant developments, and now achieves comparable quality with autoregressive (AR) models on some benchmarks, while providing an efficient alternative to AR inference. However, while AR translation is often used to implement multilingual models that benefit from transfer between languages and from improved serving efficiency, multilingual NAR models remain relatively unexplored. Taking Connectionist Temporal Classification (CTC) as an example NAR model and IMPUTER as a semi-NAR model, we present a comprehensive empirical study of multilingual NAR. We test its capabilities with respect to positive transfer between related languages and negative transfer under capacity constraints. As NAR models require distilled training sets, we carefully study the impact of bilingual versus multilingual teachers. Finally, we fit a scaling law for multilingual NAR to determine capacity bottlenecks, which quantifies its performance relative to the AR model as the model scale increases.

1 Introduction

Non-autoregressive (NAR) models generate output tokens in parallel instead of sequentially, reducing potentially expensive inference dependencies. They rely on sequence-level knowledge distillation to reach the quality of autoregressive (AR) models (Gu et al., 2018). As the notion of NAR has expanded to include semi-NAR models that generate their outputs in multiple steps, with each step generating several tokens non-autoregressively (Lee et al., 2018; Ghazvininejad et al., 2019), we have begun to see NAR matching the quality of AR (Saharia et al., 2020). Prior works have benchmarked NAR models for machine translation (MT) on a number of language pairs, but with very few exceptions, the NAR models under test have been bilingual as opposed to multilingual.

Multilingual MT models (Dong et al., 2015; Firat et al., 2017; Johnson et al., 2017), translating between multiple languages, have two major advantages. First, they offer better parameter efficiency than bilingual models via multi-tasking. Second, they are able to transfer knowledge from high-resource languages to low-resource ones. Therefore they have become an attractive solution for expanding the language coverage of AR MT (Aharoni et al., 2019; Fan et al., 2021; Siddhant et al., 2022). The capability of multilingual modeling is a major feature of the AR regime, and it is one that we should seek to maintain in NAR models.

However, it is unclear to what extent the benefits of multilingual AR models transfer to NAR modeling (Caruana, 1997; Arivazhagan et al., 2019). Do related languages help each other as easily (*positive transfer*)? Do unrelated languages interfere with one another more (*negative transfer*)? Furthermore, NAR modeling raises a new issue of multilingual distillation. To retain the training-time efficiency of multilingual modeling, it is crucial that NAR works well with multilingual teachers; otherwise, the prospect of training many bilingual teachers would greatly increase the effective training cost. It may actually be the case that multilingual teachers are better suited than bilingual ones, as the effective capacity reduction may result in less complex (Zhou et al., 2019) and less multi-modal outputs (Gu et al., 2018).

We present an empirical study of multilingual NAR modeling. Taking CTC (Libovický and Helcl, 2018) as our canonical NAR method, and IMPUTER (Saharia et al., 2020) as our canonical semi-NAR model, we study how they respond to multilinguality through a series of "stress-tests", first in a six-language scenario designed to emphasize negative transfer (§4), and then in two-language scenarios designed to emphasize positive transfer under data resource constraints (§5). Lastly, we fit a scaling law for our six-language sce-

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 177 - 187
December 7-8, 2022 ©2022 Association for Computational Linguistics

nario to measure the potential of increasing model sizes (§6). The main findings can be summarized as follows:

1. Multilingual NAR models work equally well whether datasets are distilled from bilingual or multilingual teachers.
2. Multilingual NAR models do benefit from positive transfer in scenarios that encourage it; however, in comparison to AR models, they suffer more from negative transfer and benefit less from positive transfer.
3. The scaling law demonstrates that this trend continues as model size increases.

Our extensive analysis on outputs from the NAR models suggest that they still struggle to generate "valid" tokens with desired output length. Furthermore, our results indicate that scaling up the NAR models is not going to close the gap to multilingual AR, but our analysis points to promising directions for future work throughout the paper.

2 Non-Autoregressive Multilingual NMT

Let, $D^l = (x, y) \in X \times Y$ denote the bilingual corpus of a language pair, l. Given an input sequence x of length T', an AR model (Bahdanau et al., 2015; Vaswani et al., 2017) predicts the target y with length T sequentially based on the conditional distribution $p(y_t \mid y_{<t}, x_{1:T'}; \theta)$. NAR models assume conditional independence in the output token space; that is, they model $p(y_t \mid x_{1:T'}; \phi)$. Due to this conditional independence assumption, training NAR models directly on the true target distribution leads to degraded performance (Gu et al., 2018). Hence, NAR models are typically trained with sequence-level knowledge distillation (Kim and Rush, 2016) to reduce the modeling difficulty.

2.1 Non-Autoregressive NMT with CTC

In this work, we focus on NAR modelling via CTC (Graves et al., 2006) due to its superior performance on NAR generation and the flexibility of variable length prediction (Libovický and Helcl, 2018; Saharia et al., 2020; Gu and Kong, 2021).

CTC models an alignment a that provides a mapping between a sequence of predicted and target tokens. Alignments can be constructed by inserting special *blank tokens* ("_") and token repetitions into the target sequence. The alignment is monotonic with respect to the target sequence and is always the same length as the source sequence x. However, in MT, the target sequence y can be longer than the source sequence x. This is handled via upsampling the source sequence x, to s times its original length. An alignment is valid only if when collapsed, i.e., merging repeated tokens and removing blank tokens, it results in the original target sequence. The CTC loss marginalizes over all possible valid alignments $\Gamma(y)$ compatible with the target y and is defined as:

$$p(y \mid x) = \sum_{a \in \Gamma(y)} \prod_{1 \leq t' \leq T'} p(a_{t'} \mid x_{1:T'}; \phi).$$

Note that each alignment token $a_{t'}$ is modeled independently. This conditional independence allows CTC to predict the single most likely alignment non-autoregressively at inference time, which can then be efficiently collapsed to an output sequence. This same independence assumption enables efficient minimization of the CTC loss via dynamic programming (Graves et al., 2006). While CTC enforces monotonicity between the target and the predictions, it does not require any cross- or self-attention layers inside the model to be monotonic. Hence, CTC should still be able to model language pairs with different word orders between the source and the target sequence. Following Saharia et al. (2020), we train encoder-only CTC models, using a stack of self-attention layers to map the source sequence directly to the alignments.

2.2 Iterative Decoding with Imputer

IMPUTER (Saharia et al., 2020) extends NAR CTC modeling by iterative refinement (Lee et al., 2018). At each inference step, it conditions on a previous partially generated alignment to emit a new alignment. While IMPUTER, like CTC, generates all tokens at each inference step, only a subset of these tokens is selected to generate a partial alignment, similar to iterative masking approaches (Ghazvininejad et al., 2019). This is achieved during training via marginalizing over partial alignments as follows:

$$p(y \mid x) = \sum_{a \in \Gamma(a)} p(a \mid a_{\text{Mask}}, x; \phi),$$

where a_{Mask} is a partially masked input-alignment. At training time, the a_{Mask} alignment is generated using a CTC model trained on the same dataset, and its masked positions are selected randomly. This training procedure enables IMPUTER to iteratively refine a partial alignment over multiple

	Tgt Word Order	Size	Script Difference	White Space	Avg. Src Length	Avg. Tgt Length
En-Kk	SOV	150K	✓	✓	26.7	20.0
En-De	SVO/SOV	4.6M	✗	✓	25.7	24.3
En-Pl	SVO	5M	✗	✓	16.2	14.6
En-Hi	SOV	8.6M	✓	✓	18.3	19.8
En-Ja	SOV	17.9M	✓	✗	21.4	25.9
En-Ru	Free	33.5M	✓	✓	23.2	21.5
En-Fr	SVO	38.1M	✗	✓	29.2	32.8

Table 1: Details on training data used. Target word orders are the ones that are dominating within the language according to (Dryer and Haspelmath, 2013), but there may be sentence-specific variations. English follows predominantly SVO (Subject-Verb-Object) order. Size is measured as the number of parallel sentences in the training data. Source (Src) and Target (Tgt) length are averaged across sentences after word-based tokenization.

decoding steps at inference time — consuming its own alignments as input to the next iteration. With $k > 1$ decoding steps, the IMPUTER becomes *semi-autoregressive*, requiring k times more inference passes than pure CTC models.

IMPUTER differs from Conditional Masked Language Modeling (CMLM) (Ghazvininejad et al., 2019) in that it uses the CTC loss instead of the standard cross-entropy loss, removing the need for explicit output length prediction. Also, IMPUTER is an encoder-only model that makes one prediction per source token, just like CTC. The cross-attention component from encoder-decoder is replaced by a simple sum between the embeddings of the source sequence and the input alignment (a_{Mask}) before the first self-attention layer.[1]

2.3 Multilingual Modeling

Multilingual AR and NAR models are trained on datasets from multiple language pairs, $\{D^l\}_{l=1}^{L}$. We prepend each source sequence with the desired target language tag (<2tgt>) and generate a shared vocabulary across all languages (Johnson et al., 2017). The models encode this tag as any other token, and uses it to guide the generation of the output sequence in the desired target language.

2.4 Efficiency

Inference We refrain from wallclock inference time measurements since these are dependent on implementation, low-level optimization and machines (Dehghani et al., 2021). We instead compare generation speed in terms of the number of tokens that get generated per iteration N_{gen} (Kreutzer et al., 2020), which is < 1 for AR models,[2] T for fully non-autoregressive models like CTC and $\frac{T}{k}$ for iterative semi-autoregressive models like IMPUTER. *While the potential for faster inference motivates our interest in* NAR, *our core contribution is a comparison of multilingual modeling capabilities; therefore, we do not measure inference speed experimentally.*

Training At training time, NAR models are less efficient than AR models because their quality depends on distillation (Gu and Kong, 2021). Extra cost is incurred to train a teacher model (usually AR) and to use it to decode the training set.

Multilinguality Multilingual models multi-task over language pairs, so that a single multilingual model can replace several bilingual models. Thanks to transfer across languages, model size needs to be increased less than m-fold for modeling m language pairs.

Considering all of the above factors, an ideal model needs only a few iterations (decoder passes or steps), requires no teacher or a cheap teacher, and covers several languages, while incurring the smallest drop in quality compared to less efficient models. CTC is desirable as it uses only one pass, while IMPUTER gives up some efficiency to improve quality. Both require a teacher, but we can try to reduce the cost by training fewer teachers.

3 Experimental Setup

Data We perform our main experiments on six language pairs, translating from English into WMT-14 German (De) (Bojar et al., 2014), WMT-15 French (Fr) (Bojar et al., 2015), WMT-19 Russian (Ru) (Barrault et al., 2019), WMT-20 Japanese (Ja), WMT-20 Polish (Pl) (Barrault et al., 2020) and Samanantar Hindi (Hi) (Ramesh et al., 2021). The lower-resourced WMT-19 English-Kazakh (Kk) (Barrault et al., 2019) is used for an additional transfer experiment in Section 5. The properties

[1] We experimented with an encoder-decoder variant of IMPUTER but it did not change the overall output quality in multilingual scenarios or otherwise.

[2] 1 for greedy search, < 1 to account for scoring and expansion of multiple hypotheses in beam search.

of the datasets are listed in Table 1. Target word order and writing script notably differ across these languages, so we focus on translating *into* these languages as this is a more challenging direction. A shared sub-word vocabulary of 32k is trained with SentencePiece (Kudo and Richardson, 2018), with the number of sub-words allocated for each language being proportional to its data size.

Evaluation Metrics Translation quality is evaluated with BLEU (Papineni et al., 2002) as calculated by Sacrebleu (Post, 2018) with default tokenization ("13a") except for EN-JA, where we use character-level tokenization. [3]

Architecture We train the IMPUTER model using the same setup as described in Saharia et al. (2020): We follow their base model with d_{model} = 512, d_{hidden} = 2048, n_{heads} = 8, n_{layers} = 12, and $p_{dropout}$ = 0.1. AR models follow Transformer-base (Vaswani et al., 2017) and have similar parameter counts. We train both models using Adam with learning rate of 0.0001. We train CTC models with a batch size of 2048 and 8192 sentences for 300K steps for the bilingual and multilingual models respectively. We train the IMPUTER using CTC loss using a Bernoulli masking policy for next 300K steps with a batch size of 1024 and 2048 sentences for the bilingual and multilingual models respectively. We upsample the source sequence by a factor of 2 for all our experiments.[4] We pick the best checkpoint based on validation BLEU for bilingual models, and the last checkpoint for multilingual models, following Arivazhagan et al. (2019).

Distillation We apply sequence-level knowledge distillation (Kim and Rush, 2016) from AR teacher models as widely used in NAR generation (Gu et al., 2018). Specifically, when training the NAR models, we replace the reference sequences during training with translation outputs from Transformer-Big AR teacher model with a beam width of four. We also report the quality of the AR teacher models, both bilingual and multilingual. The configurations for training the big AR teacher models also follow Vaswani et al. (2017).

4 Negative Transfer Scenario

Our main experiment compares bilingual, multilingual, AR and NAR models for the six high-resource languages from Table 1. These languages are typologically diverse, and they each have enough data so that we do not expect them to benefit substantially from joint modeling. We use this challenging scenario to test the impact of multilingual teachers, and to measure each paradigm's ability to model several unrelated languages. Results are shown in Table 2.

4.1 Multilingual Teacher Comparison

Inspecting the AR teacher models (rows 1 and 2 of Table 2) confirms the negative transfer that we aimed to design: multilingual teachers have substantially reduced BLEU compared to bilingual teachers. How much is this drop in quality affecting NAR students? First of all, we see that bilingual CTC models trained from the multilingual teacher (5) do not reflect the entirety of this drop when compared to training with the bilingual teacher (4): An average teacher gap of −1.8 BLEU is causing −1.1 drop for the corresponding students.[5] The comparison becomes more interesting as we shift to multilingual students: multilingual CTC (8, 9) does not suffer at all from having a multilingual teacher (average BLEU gap of −0.1), and multilingual IMPUTER (10, 11) likewise suffers very little (−0.3). These three results taken together suggest that *datasets distilled from multilingual models are likely simpler, but easier to model non-autoregressively* by the multilingual NAR models, which makes up for the teacher's lower BLEU. Our analysis in Section 4.3 supports this hypothesis.

We hope that highly multilingual models, trained with similar target language pairs to enhance positive transfer (Tan et al., 2019), are even better suited to serve as teachers for multilingual NAR models, which we leave to future work.

4.2 Multilingual Student Comparison

Returning to the "Bilingual Models" section of Table 2 with `AR-big` teachers, we can see that we have reproduced the results of Saharia et al. (2020): Bilingual CTC (4) performs well for a fully NAR method, but does not reach AR quality (3). IMPUTER (6) ably closes the gap with AR, surpassing

[3] SacreBLEU (short) signatures: nrefs:1|case:mixed|eff:no|tok:13a,char|smooth:exp|version:2.0.0

[4] We do not vary the upsampling ratio due to small difference in the performance of the resulting NAR models (see Table 6, Gu and Kong (2021)).

[5] As the quality of CTC generated alignments from the multi-AR-big is worse than the alignments generated from the CTC with the AR-big teacher, we do not train IMPUTER on CTC-generated alignments from the multi-AR-big models.

	Model	Teacher	N_{gen}	En-Fr	En-De	En-Pl	En-Ru	En-Hi	En-Ja	Avg.
	Teachers									
(1)	AR-big		< 1	38.8	29.0	21.4	27.2	34.6	35.4	31.1
(2)	multi-AR-big			38.5	27.0	21.6	25.3	32.6	33.6	29.3
	Bilingual Models									
(3)	AR-base		< 1	38.2	27.6	21.2	26.2	33.8	34.8	30.3
(4)	CTC	AR-big	T	35.7	25.2	18.0	21.4	31.6	31.6	27.3
(5)		multi-AR-big		35.1	24.0	17.7	20.8	30.8	28.9	26.2
(6)	Imputer	AR-big	$\frac{T}{8}$	38.5	27.2	21.2	25.6	32.0	32.0	29.4
	Multilingual Models									
(7)	multi-AR-base		< 1	35.2	24.8	19.7	23.2	30.8	31.2	27.5
(8)	CTC	AR-big	T	31.6	20.5	13.0	17.7	28.2	28.1	23.2
(9)		multi-AR-big		31.2	20.5	13.7	18.0	27.8	27.5	23.1
(10)	Imputer	AR-big	$\frac{T}{8}$	34.4	22.8	14.9	21.3	29.9	29.6	25.5
(11)		multi-AR-big		34.1	21.2	16.4	21.7	29.9	27.9	25.2

Table 2: Test BLEU scores for multilingual and bilingual AR and NAR models and their teachers.

or coming within 0.4 BLEU of the AR-base models on 3/6 language pairs, with the largest gap in performance for the distant En-Ja. Does this story hold as we move to multilingual NAR students?

To understand each model's multilingual capabilities, we can compare its bilingual performance to its multilingual performance. Comparing bilingual AR-base (3) to its multilingual counterparts (7) gives us a baseline average drop of −2.8 BLEU, confirming that this is indeed a difficult multilingual scenario that leads to negative transfer. Comparing bilingual CTC (4) to multilingual CTC (8) with AR-big teachers, we see an average drop of −4.1. This larger drop indicates that CTC *suffers more from negative interference than its* AR *counterpart*. We hypothesize that CTC models need more capacity than AR models to achieve similar multilingual performance, motivating our scaling law experiments in Section 6.

Performing the same bilingual-to-multilingual comparison for Imputer (6 vs. 10) shows a similar −3.9 average drop due to negative transfer. So although Imputer is indeed better than CTC (2 BLEU), it does not seem to be better suited for multilingual modeling in this difficult scenario.

4.3 How do the bilingual and the multilingual distilled datasets differ?

Table 3 summarizes different statistics for the original (R) and distilled datasets from both multilingual (M) and bilingual (B) AR teacher models. We report the number of types and average sequence length (in tokens) for the target side of the dataset. We compute the complexity of the dataset based on probabilities from a statistical word aligner (Zhou et al., 2019). The FRS (Talbot et al., 2011) score represents the average fuzzy reordering score over all the sentence pairs for the respective language pair as measured in Xu et al. (2021), with higher values suggesting that the target is more monotonic with the source sequence. We also report BLEU for the distilled datasets relative to the original training references.

The datasets distilled from the bilingual AR models (B) are shorter, less complex, have reduced lexical diversity (in number of types) and are more monotonic compared to the original corpora (R), which corroborates findings from prior work (Zhou et al., 2019; Xu et al., 2021). One exception is En-Ja, where the distilled translations are slightly less monotonic than the original references. Moving to multilingual teachers (M), the resulting datasets have further reduced types, are shorter and less complex than those distilled from bilingual teachers. In particular, their monotonicity increased (FRS) for the more distant language pairs, En-Ja and En-Hi. As shown in Xu et al. (2021) and Voita et al. (2021), reduced lexical diversity and reordering complexity can help bilingual NAR models to learn better alignments between source and target, improving the translation quality of the outputs. More work is needed to better understand

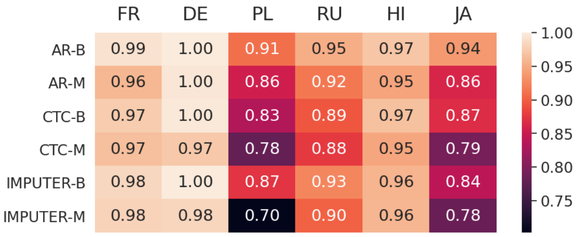
PROPERTY	R	B	M
EN-FR			
# TYPES	522K	430K	396K
AVG. LENGTH	32.8	31.2	29.2
COMPLEXITY	1.529	1.167	0.944
FRS	0.463	0.541	0.536
BLEU (Train)	-	40.8	37.8
EN-DE			
# TYPES	812K	616K	573K
AVG. LENGTH	24.3	23.4	22.2
COMPLEXITY	1.243	0.819	0.709
FRS	0.490	0.606	0.605
BLEU (Train)	-	35.0	26.4
EN-PL			
# TYPES	636K	516K	503K
AVG. LENGTH	14.6	13.4	12.7
COMPLEXITY	1.435	0.942	0.591
FRS	0.590	0.678	0.695
BLEU (Train)	-	26.3	22.0
EN-RU			
# TYPES	636K	516K	503K
AVG. LENGTH	21.5	20.5	19.5
COMPLEXITY	1.083	0.882	0.819
FRS	0.640	0.719	0.716
BLEU (Train)	-	43.2	40.0
EN-HI			
# TYPES	346K	200K	185K
AVG. LENGTH	19.8	18.8	17.8
COMPLEXITY	1.438	1.256	1.138
FRS	0.347	0.363	0.366
BLEU (Train)	-	34.6	28.0
EN-JA			
# TYPES	547K	440K	402K
AVG. LENGTH	25.9	23.5	22.2
COMPLEXITY	1.541	1.369	1.338
FRS	0.344	0.337	0.340
BLEU (Train)	-	35.9	30.6

Table 3: Comparison of datasets (1M samples) distilled from bilingual (B) or multilingual (M) AR models

the sweet-spot between the quality and complexity trade-off of the multilingual and bilingual distilled datasets for multilingual NAR modeling.

4.4 Which translation errors are made?

In this section, we analyze quantitatively how the output quality of NAR models differs across language pairs when trained in isolation (bilingual) or with other language pairs (multilingual).



	FR	DE	PL	RU	HI	JA
AR-B	0.99	1.00	0.91	0.95	0.97	0.94
AR-M	0.96	1.00	0.86	0.92	0.95	0.86
CTC-B	0.97	1.00	0.83	0.89	0.97	0.87
CTC-M	0.97	0.97	0.78	0.88	0.95	0.79
IMPUTER-B	0.98	1.00	0.87	0.93	0.96	0.84
IMPUTER-M	0.98	0.98	0.70	0.90	0.96	0.78

Figure 1: Brevity penalty scores for bilingual (-B) and multilingual (-M) models, the closer to 1 the better.

Effect of Length Figure 1 shows the brevity penalty (BP) scores (Papineni et al., 2002) for all languages. EN-PL and EN-JA have lowest BP scores across the board, meaning that their translations are shorter than the references. Manual inspection reveals that this could be attributed to the subject pronouns being dropped in both of these target languages. Multilingual modeling results in shorter outputs relative to bilingual models for both AR and NAR models and most language pairs. While IMPUTER models tend to have fewer issues with output length compared to CTC models, they still lag behind AR models, suggesting that the length might need to be controlled explicitly for these language pairs (Gu and Kong, 2021).

Invalid Words CTC frequently generates *invalid* words, i.e. tokens that are not present in the target side of the bitext but are being composed from multiple sub-words. These sub-words represent alternative translations that the model fails to distinguish. In the Hindi example below, the invalid (or made-up) word in the sentence is marked in red. The correct word should be जहरीले as the dependent vowel "ी" can only be used once.

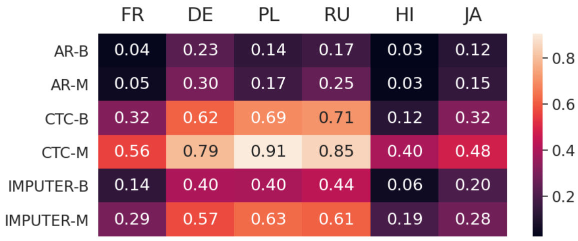
Hindi: इससे ग्रामीण महिलाओं को जहरीीले धुएं से मुक्ति मिली हैं।

English: This has relieved the rural women from the poisonous smoke.

Figure 2 reports the percentage of sequences that include at least one invalid word in the test set. CTC generates many invalid words compared to both AR and IMPUTER, with multilingual modeling leading to an average increase in invalid words by 37%. The shared vocabulary of the multilingual model results in shorter sub-words, hence longer sequences, and the conditionally independent generation leads to more clashing adjacent sub-words.[6]

[6] One might hope to alleviate this by increasing vocabulary size, but preliminary experiments showed that an increased vocabulary was less efficient in improving quality than increasing overall model size, which is explored in Section 6.

IMPUTER's iterative decoding alleviates this for some languages. Increasing the number of iterations could help, but would also erode the efficiency arguments that make NAR models attractive. As pointed out by Xiao et al. (2022), better modeling of target token dependencies is crucial to closing the gap in translation quality to AR models.



	FR	DE	PL	RU	HI	JA
AR-B	0.04	0.23	0.14	0.17	0.03	0.12
AR-M	0.05	0.30	0.17	0.25	0.03	0.15
CTC-B	0.32	0.62	0.69	0.71	0.12	0.32
CTC-M	0.56	0.79	0.91	0.85	0.40	0.48
IMPUTER-B	0.14	0.40	0.40	0.44	0.06	0.20
IMPUTER-M	0.29	0.57	0.63	0.61	0.19	0.28

Figure 2: % of outputs with invalid words for bilingual (-B) and multilingual (-M) models, the lower the better.

5 Positive Transfer Scenario

In this section we present two experimental setups designed to emphasize positive transfer, where languages are related and training data is limited.

English→{German, French} To isolate the effect of transfer via multilingual modelling, we relax the capacity bottleneck and competition for parameters: We combine the two most related languages (DE, FR) (Kudugunta et al., 2019, Figure 2) and give them smaller, balanced training sets (1M sentences). We compare bilingual and multilingual AR and NAR models trained on this reduced data.

Table 4 shows that NAR *models benefit from training with multiple language pairs in this relaxed scenario* — all models exhibit positive transfer (in green). IMPUTER achieves higher positive transfer than CTC for both languages, but lags behind the AR multilingual model in EN-FR. However, for EN-FR the bilingual IMPUTER is already ahead of the bilingual AR model by 0.4 BLEU.

MODEL	EN-DE	EN-FR
Bilingual Models		
AR	22.8	27.7
CTC	21.5	26.5
IMPUTER	22.8	28.1
Multilingual Models		
AR	**24.3** +1.5	**29.0** +1.3
CTC	22.1 +0.6	26.9 +0.4
IMPUTER	23.7 +1.3	28.5 +0.4

Table 4: Results on subsampled (1M) training data.

English→{Russian, Kazakh} Does this positive transfer survive data imbalance? We test the performance of the multilingual NAR model on the low-resource task of translating English into Kazakh, for which the size of clean training data is insufficient to train a bilingual AR model from scratch. We instead distill translations from the publicly available multilingual AR model, PRISM (Thompson and Post, 2020). We then pair it with the higher-resource but related language Russian to encourage positive transfer to Kazakh. Given the huge difference in data sizes for Russian and Kazakh (see Table 1), we sample training data from the two languages based on the data size scaled by a temperature value τ, $p_l^{1/\tau}$ (Arivazhagan et al., 2019), where, $p_l = \frac{D_l}{\sum_k D_k}$. We experiment with multiple temperature values (1, 3, 5, 10, 20) and pick the best value ($\tau = 5; p_{\text{RU}}^{1/\tau} = 0.75, p_{\text{KK}}^{1/\tau} = 0.25$) based on the performance on the validation set.

MODEL	TEACHER	EN-KK	EN-RU
PRISM	-	**8.9**	**27.0**
Bilingual Models			
AR	PRISM	4.4	-
CTC		1.2	-
Multilingual Models			
AR	PRISM	7.1 +2.7	26.0
CTC		2.8 +1.6	20.4

Table 5: Results on English → Kazakh, Russian.

As can be seen in Table 5, both AR and CTC show positive transfer when translating into Kazakh when trained in combination with Russian. The multilingual CTC model is able to improve over the bilingual CTC model, but the overall quality of the outputs is very low compared to the teacher model (BLEU: -5.3). This experiment showcases that *current* NAR *models do not perform well on very low-resource language pairs* and might need further data augmentation (Song et al., 2022) or transfer from other similar languages.[7]

6 Impact of Model Scale

We hypothesized in Section 4 that CTC might require more capacity than AR models. If we increase the parameters for NAR models sufficiently, could we reach AR quality? Scaling laws can characterize the relationship between MT quality, the cross-entropy loss and the number of parameters

[7] We do not train IMPUTER for KK as the quality of the distilled dataset and alignments from CTC is very low.

used for training the model (Ghorbani et al., 2021; Gordon et al., 2021).

We derive the relationship between BLEU and the number of parameters for our AR and CTC models directly from the scaling laws proposed by Gordon et al. (2021) and Ghorbani et al. (2021) as follows:

$$\begin{aligned} L(N) &\approx L_0 + \alpha_n (1/N)^{\alpha_k} \text{ (Ghorbani et al., 2021)} \\ \mathrm{BLEU}(L) &\approx C\mathrm{e}^{-kL} \text{ (Gordon et al., 2021)} \\ \mathrm{BLEU}(N) &\approx a\mathrm{e}^{-b(1/N)^c} \text{ (this work)} \end{aligned}$$

where L is the test loss, $\{\alpha_n, \alpha_k, L_0, C, k\}$ are fitted parameters from previous power laws, and $\{a, b, c\}$ are the collapsed fitted parameters of our power law. Ghorbani et al. (2021)'s L_0 corresponds to the irreducible loss of the data (here: a).

Setup We train seven models with varying capacity for AR and CTC models. The number of layers and model sizes are varied as: (6, 128), (6, 256), (12, 256), *(12, 512)*,[8] (24, 512), (12, 1024), (24, 1024). The feed-forward size is 4× the model size. AR models have equal numbers of encoder and decoder layers. The number of attention heads is given by $(8/(512/\text{Model Size}))$. For a fair comparison, we train both AR and CTC models on distilled outputs from `AR-big` in Table 2. The evaluation is conducted in the challenging six-language negative-transfer scenario from Section 4, where capacity bottlenecks are likely to be most pronounced. We report BLEU averaged across six languages.

Results Figure 3 shows the fitted parameters using the scaling law, which can almost perfectly describe the relationship between the number of parameters and the development BLEU (R^2: 0.99). We can see that CTC*, even with many more parameters, do not come even close to the performance of* AR *models and plateaus early* at a BLEU of 26.7, while AR models plateau at 30.8. By projecting the curves out to 1 billion parameters, we show that increasing the capacity of NAR is insufficient to reach the quality of AR models.

7 Related Work

Multiple approaches with varying architectures (Gu et al., 2018, 2019; Chan et al., 2020; Xu and Carpuat, 2021), custom loss functions (Ghazvininejad et al., 2020; Du et al., 2021) and training strategies (Ghazvininejad et al., 2019; Qian et al., 2021) have been used to enable parallel generation of output tokens for MT with sequence-level knowledge distillation as one of the key ingredient in the training of NAR models. Both supervised, and unsupervised (Sun et al., 2020) MT have benefitted from training with multiple languages, especially those that have tiny (Siddhant et al., 2020) to no training data (Zhang et al., 2020). However, multilingual modeling has not yet received any attention in the NAR literature, which we explore in this work. One limitation of our study is that we choose one representative system for NAR and semi-NAR modeling over the full breadth of NAR options.

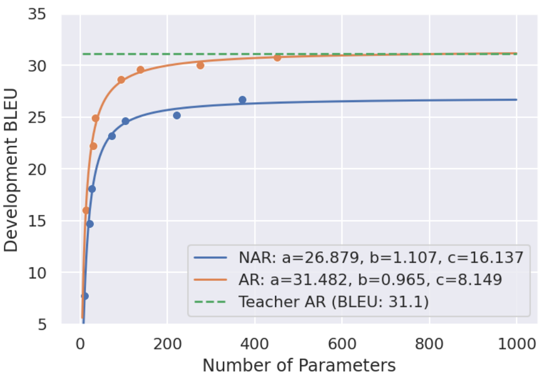


Figure 3: BLEU versus number of parameters and fitted power-law curves (R^2 AR: 0.99, R^2 CTC: 0.99).

8 Conclusion

Multilingual translation is a valuable feature of AR models, therefore, we have tested NAR models for that same capability. We focus on challenging scenarios to discover potential weaknesses and to identify areas for future work. In a relaxed setting with little interference between languages and balanced data, multilingual NAR models nicely exhibit positive transfer, practically closing the gap to AR models with a few decoding iterations. However, we do not see the same positive transfer in a true low-resource scenario. Experiments in a six-language scenario reveal that multilingual NAR models suffer proportionally more from negative interference than AR models. Our derived scaling laws show that scaling up CTC model parameters is not a sufficient remedy. Our analysis identified two issues that hurt translation quality and worsen with multilinguality, namely output length control and the generation of invalid words. We have also shown beneficial properties of using multilingual teachers for distillation. We hope that this work will serve as a call for increased focus on multilingual modeling in NAR research.

[8] Size for experiments in Section 4.

Acknowledgments

We thank George Foster, Aditya Gupta, and the anonymous reviewers for their helpful and constructive comments.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of NAACL-HLT*, pages 3874–3884.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R Costa-Jussa, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, et al. 2019. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation*, pages 12–58.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

William Chan, Chitwan Saharia, Geoffrey Hinton, Mohammad Norouzi, and Navdeep Jaitly. 2020. Imputer: Sequence modelling via imputation and dynamic programming. In *International Conference on Machine Learning*, pages 1403–1413. PMLR.

Mostafa Dehghani, Anurag Arnab, Lucas Beyer, Ashish Vaswani, and Yi Tay. 2021. The efficiency misnomer. *arXiv preprint arXiv:2110.12894*.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Cunxiao Du, Zhaopeng Tu, and Jing Jiang. 2021. Order-agnostic cross entropy for non-autoregressive machine translation. *arXiv preprint arXiv:2106.05093*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Orhan Firat, Kyunghyun Cho, Baskaran Sankaran, Fatos T. Yarman Vural, and Yoshua Bengio. 2017. Multi-way, multilingual neural machine translation. *Computer Speech & Language*, 45:236–252.

Marjan Ghazvininejad, Vladimir Karpukhin, Luke Zettlemoyer, and Omer Levy. 2020. Aligned cross entropy for non-autoregressive machine translation. *CoRR*, abs/2004.01655.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121.

Behrooz Ghorbani, Orhan Firat, Markus Freitag, Ankur Bapna, Maxim Krikun, Xavier Garcia, Ciprian Chelba, and Colin Cherry. 2021. Scaling laws for neural machine translation. *arXiv preprint arXiv:2109.07740*.

Mitchell A Gordon, Kevin Duh, and Jared Kaplan. 2021. Data and parameter scaling laws for neural machine translation. In *Proceedings of the 2021*

Conference on Empirical Methods in Natural Language Processing, pages 5915–5922.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Jiatao Gu and Xiang Kong. 2021. Fully non-autoregressive neural machine translation: Tricks of the trade. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 120–133, Online. Association for Computational Linguistics.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 11179–11189. Curran Associates, Inc.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Julia Kreutzer, George Foster, and Colin Cherry. 2020. Inference strategies for machine translation with conditional masking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5774–5782, Online. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP (Demonstration)*.

Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual NMT representations at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1565–1575, Hong Kong, China. Association for Computational Linguistics.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium. Association for Computational Linguistics.

Jindřich Libovický and Jindřich Helcl. 2018. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3016–3021, Brussels, Belgium. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2021. Glancing transformer for non-autoregressive neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1993–2003, Online. Association for Computational Linguistics.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Divyanshu Kakwani, Navneet Kumar, et al. 2021. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. *arXiv preprint arXiv:2104.05596*.

Chitwan Saharia, William Chan, Saurabh Saxena, and Mohammad Norouzi. 2020. Non-autoregressive machine translation with latent alignments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1098–1108.

Aditya Siddhant, Ankur Bapna, Yuan Cao, Orhan Firat, Mia Chen, Sneha Kudugunta, Naveen Arivazhagan, and Yonghui Wu. 2020. Leveraging monolingual data with self-supervision for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2827–2835, Online. Association for Computational Linguistics.

Aditya Siddhant, Ankur Bapna, Orhan Firat, Yuan Cao, Mia Xu Chen, Isaac Caswell, and Xavier Garcia.

2022. Towards the next 1000 languages in multilingual machine translation: Exploring the synergy between supervised and self-supervised learning. *CoRR*, abs/2201.03110.

Zhenqiao Song, Hao Zhou, Lihua Qian, Jingjing Xu, Shanbo Cheng, Mingxuan Wang, and Lei Li. 2022. switch-GLAT: Multilingual parallel machine translation via code-switch decoder. In *International Conference on Learning Representations*.

Haipeng Sun, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2020. Knowledge distillation for multilingual unsupervised neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3525–3535, Online. Association for Computational Linguistics.

David Talbot, Hideto Kazawa, Hiroshi Ichikawa, Jason Katz-Brown, Masakazu Seno, and Franz Josef Och. 2011. A lightweight evaluation framework for machine translation reordering. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 12–21.

Xu Tan, Jiale Chen, Di He, Yingce Xia, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with language clustering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 963–973.

Brian Thompson and Matt Post. 2020. Paraphrase generation as zero-shot multilingual translation: Disentangling semantic similarity from lexical and syntactic diversity. In *Proceedings of the Fifth Conference on Machine Translation*, pages 561–570, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Elena Voita, Rico Sennrich, and Ivan Titov. 2021. Language modeling, lexical translation, reordering: The training process of NMT through the lens of classical SMT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8478–8491, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yisheng Xiao, Lijun Wu, Junliang Guo, Juntao Li, Min Zhang, Tao Qin, and Tie-yan Liu. 2022. A survey on non-autoregressive generation for neural machine translation and beyond. *arXiv preprint arXiv:2204.09269*.

Weijia Xu and Marine Carpuat. 2021. Editor: An edit-based transformer with repositioning for neural machine translation with soft lexical constraints. *Transactions of the Association for Computational Linguistics*, 9:311–328.

Weijia Xu, Shuming Ma, Dongdong Zhang, and Marine Carpuat. 2021. How does distilled data complexity impact the quality and confidence of non-autoregressive machine translation? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4392–4400, Online. Association for Computational Linguistics.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2019. Understanding knowledge distillation in non-autoregressive machine translation. In *International Conference on Learning Representations*.

Learning an Artificial Language for Knowledge-Sharing in Multilingual Translation

Danni Liu and **Jan Niehues**
Karlsruhe Institute of Technology
{danni.liu, jan.niehues}@kit.edu

Abstract

The cornerstone of multilingual neural translation is shared representations across languages. Given the theoretically infinite representation power of neural networks, semantically identical sentences are likely represented differently. While representing sentences in the *continuous* latent space ensures expressiveness, it introduces the risk of capturing of irrelevant features which hinders the learning of a common representation. In this work, we *discretize* the encoder output latent space of multilingual models by assigning encoder states to entries in a codebook, which in effect represents source sentences in a new artificial language. This discretization process not only offers a new way to interpret the otherwise black-box model representations, but, more importantly, gives potential for increasing robustness in unseen testing conditions. We validate our approach on large-scale experiments with realistic data volumes and domains. When tested in zero-shot conditions, our approach is competitive with two strong alternatives from the literature. We also use the learned artificial language to analyze model behavior, and discover that using a similar bridge language increases knowledge-sharing among the remaining languages.

1 Introduction

A promising potential of multilingual (Dong et al., 2015; Firat et al., 2016; Ha et al., 2016; Johnson et al., 2017) neural machine translation (NMT) is knowledge-sharing between languages. To enable knowledge-sharing, a prerequisite is the ability to capture common features of languages, especially between related ones. *Constructed languages* such as *Interlingua* and *Esperanto* are excellent examples of human-designed structures based on the commonalities of a wide range of related languages. For data-driven models, however, it is difficult to leverage such resources due to data scarcity: There is little parallel data to these constructed languages, and creating new translation heavily depends on expert curation. Instead of relying on manually-created data, we aim to learn an artificial language in a more unsupervised fashion in parallel with training the NMT model. Specifically, our goal is to learn a sequence of tokens to represent the source sentences, which then serves as context for the NMT decoder. Table 1 illustrates this idea.

source sentence (English)	learning	a	new	language
	↓	↓	↓	↓
discrete codes	3	609	57	1042
source sentence (Indonesian)	belajar	bahasa	baru	
	↓	↓	↓	
discrete codes	3	57	258	

Table 1: We aim to learn a sequence of discrete codes to represent source sentences in multilingual NMT models. Our goal is to 1) improve inference-time robustness, 2) have more interpretable intermediate representations.

A potential advantage of representing inputs in discrete tokens is *robustness*, a property especially relevant when NMT systems must cope with unexpected testing conditions. By discretization, we restrict the continuous latent space to a finite size, providing the possibility for model intermediate representations to fall back to a position seen in training. For instance, in zero-shot translation, where the model translates directions never seen in training, the inference-time behavior is often unstable (Gu et al., 2019; Al-Shedivat and Parikh, 2019; Rios et al., 2020; Raganato et al., 2021). In practice, pivoting through an intermediate language typically gives a strong performance upper bound difficult to surpass by direct zero-shot translation (Al-Shedivat and Parikh, 2019; Arivazhagan et al., 2019a; Zhu et al., 2020; Yang et al., 2021b). Mapping the source sentences to discrete codes could act as a *pseudo*-pivoting step, which we hope to make the model more robust under zero-shot conditions.

The discrete codes also provide a new way to interpret model representations. While there are a

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 188 - 202
December 7-8, 2022 ©2022 Association for Computational Linguistics

wealth of methods to analyze knowledge-sharing in multilingual NMT (Aji et al., 2020; Mueller et al., 2020; Chiang et al., 2022), they mostly either measure translation performance as a proxy, or involve sophisticated post-processing after model training, e.g. correlation scores between model hidden states (Kudugunta et al., 2019; Chiang et al., 2022), training classifiers to probe linguistic features (Liu et al., 2021a), or pruning model submodules (Kim et al., 2021). In contrast, when the model hidden states are directly associated with discrete tokens, they are directly more *interpretable*. This characteristic is especially relevant in unseen testing conditions, where it is important to pinpoint the underlying cause of model behavior.

Despite the advantages, discretizing the latent space of NMT models makes them inherently less expressive than their fully continuous counterparts. Maintaining translation performance relative to the continuous models is therefore a challenge. To strike a balance between expressiveness and discretization, we propose a *soft* discretization approach: In training, we assign each encoder hidden state to an entry in a fixed-size codebook. This step in effect clusters encoder hidden states to one of the many cluster centers in the latent space. The codebook where the cluster centers come from is then trained along with the translation model. To ensure that the decoder receives sufficient context information, we make it access both the discretized or continuous context, as illustrated in Figure 1. In our experiments on data from the Large-Scale Multilingual Translation Shared Task (Wenzek et al., 2021) from WMT21 (Akhbardeh et al., 2021), our approach is able to learn meaningful discrete codes and achieve translation performance competitive with models with continuous latent spaces. Our main contributions are:

- We propose a framework to learn discrete tokens as intermediate representations of multilingual NMT models (§3).
- On large-scale multilingual translation experiments, our approach is competitive with strong alternatives while offering more interpretable intermediate representations (§5.1).
- We use the learned discrete codes to study the role of bridging languages. Using two novel analyses, namely *code overlap* and *code translation*, we discover that using a similar bridge language facilitates knowledge-sharing in all languages covered by the model (§5.2).

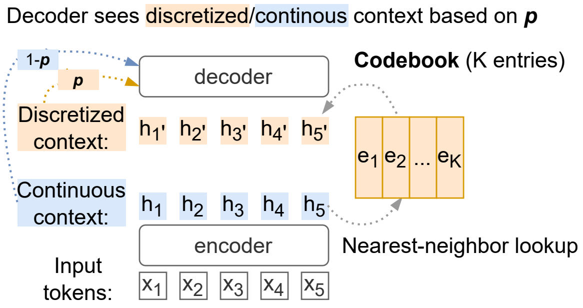

Decoder sees discretized/continous context based on *p*
1-*p*
p
decoder
Codebook (K entries)
Discretized context:
h_1' h_2' h_3' h_4' h_5'
e_1 e_2 ... e_K
Continuous context:
h_1 h_2 h_3 h_4 h_5
encoder
Nearest-neighbor lookup
Input tokens:
x_1 x_2 x_3 x_4 x_5

Figure 1: An illustration of our approach, which introduces a codebook for discretizing the encoder output latent space. During training, the decoder sees discretized and continuous context based on probability p. For inference, we use the continuous context, which have been well-clustered into a set of cluster centers after training.

2 Related Work

Multilingual Machine Translation Multilingual translation models are able to multitask over many language pairs. For this large-scale multi-task learning problem, training data plays a critical role. Low-resource directions often need upsampling to perform well (Arivazhagan et al., 2019b; Tang et al., 2021), which, meanwhile, brings capacity bottlenecks (Aharoni et al., 2019) to high-resource languages. This capacity bottleneck can be eliminated by dedicated language-specific capacity (Bapna and Firat, 2019; Philip et al., 2020; Shazeer et al., 2017; Zhang et al., 2021). When scaling up translation coverage (Aharoni et al., 2019; Zhang et al., 2020; Fan et al., 2021), zero-shot directions that have not seen any parallel training data is more likely to get encountered. While many dedicated models or objectives have been proposed to improve the zero-shot performance (Al-Shedivat and Parikh, 2019; Arivazhagan et al., 2019a; Pham et al., 2019; Zhu et al., 2020; Son and Lyu, 2020; Liu et al., 2021a; Yang et al., 2021b; Raganato et al., 2021), there is in general a *tradeoff* between supervised and zero-shot performance.

Robustness in Zero-Shot Conditions Zero-shot generalization is a widely-discussed direction in machine learning research (Socher et al., 2013; Norouzi et al., 2014; Romera-Paredes and Torr, 2015; Xian et al., 2017). In the context of NMT, early multilingual models already possess some capability of zero-shot translation of directions unseen in training (Ha et al., 2016; Johnson et al., 2017). However, zero-shot performance has been shown highly sensitive to, among other factors, training data diversity (Rios et al., 2020), language token strategies (Wu et al., 2021; ElNokrashy et al., 2022), and dropout configurations (Arivazhagan et al., 2019a; Liu et al., 2021b). A main cause of the degraded quality is that the zero-shot inference generates *off-target* translation (Zhang et al., 2020) into a language other than the desired one. In recent shared tasks (Anastasopoulos et al., 2021; Libovický and Fraser, 2021a), generating synthetic data by back-translation (Sennrich et al., 2016) to eliminate zero-shot conditions has been a dominant approach for improving upon pure unsupervised settings (Pham et al., 2021; Zhang and Sennrich, 2021; Liu and Niehues, 2021; Knowles and Larkin, 2021; Libovický and Fraser, 2021b). A main motivating factor for converting zero-shot conditions to semi-supervised ones is that the latter provides more robust and consistent inference-time behavior. In this light, to fully realize the potential of knowledge-sharing in multilingual NMT, improving zero-shot robustness is an essential task.

Discrete Representations Vector Quantized Variational Autoencoder (VQ-VAE; van den Oord et al. 2017) learns discrete tokens for continuous inputs such as images and audio, and showed its effectiveness in creating discrete representations for speech representations on practical tasks (Tjandra et al., 2020; Baevski et al., 2020). Kaiser et al. (2018) proposed an improvement to VQ-VAE by *slicing*, i.e. decomposing to quantization input and output into several subspaces. The sliced variant was used in auto-encoding for learning shorter sequences, which allows to accelerate the target generation in auto-regressive decoders. The most related work to ours is probably that of Escolano et al. (2021), who used sliced VQ-VAE (Kaiser et al., 2018) on translation tasks. The main difference is that our focus is fully parameter shared multilingual systems while Escolano et al. (2021) focused on auto-encoding and bilingual systems using language-specific encoders and decoders. Therefore, in Escolano et al. (2021) zero-shot translation only occurs after a subsequent training step on dedicated encoder for the new language. Moreover, our approach extends sliced VQ-VAE (Kaiser et al., 2018) by soft codes that utilizes both continuous and quantized encoder hidden states.

3 Learning Discrete Codes

As motivated in §1, we aim to learn to represent sources sentences with a sequence of discrete codes out of a codebook. To this end, alongisde the translation objective, we also train our model to partition the continuous latent space of the encoder output into discrete subspaces. Each of the discrete subspaces is represented by one of the k entries (cluster centers) from a trainable codebook, and the encoder hidden states are assigned to these entries. To learn a meaningful discretization, the learned cluster centers must fulfill some requirements: 1) avoid trivial solutions where all points are assigned to one or a few codebook entries, 2) carry sufficient context information for the decoder for the translation task, despite being less expressive than the encoder output prior to the discretization step.

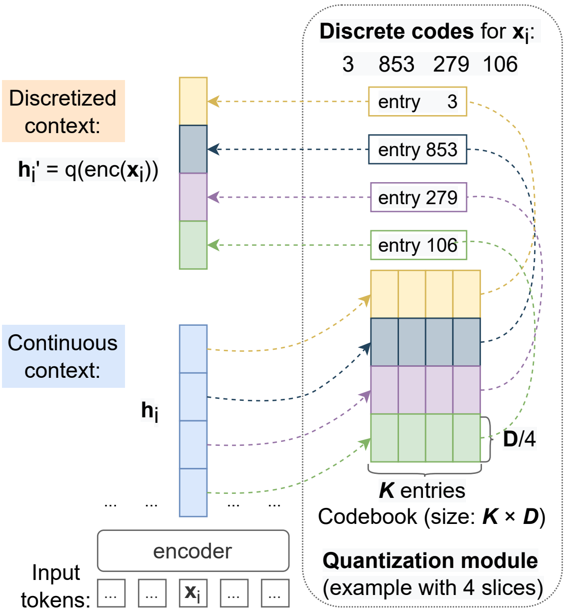


Figure 2: Illustration of the generation of the discrete codes based on a sliced (Kaiser et al., 2018) codebook.

3.1 Discretizing Encoder Latent Space

Compared to a standard Transformer (Vaswani et al., 2017), our model includes a quantization module between the encoder and decoder. We denote the quantization operation as $q(\cdot)$. Before being passed to the decoder, the encoder hidden states $\text{enc}(X)$ for input sequence X first goes through the quantization module, which runs a nearest neighbor lookup in an embedding table, i.e. the codebook. Following the notations from van den Oord et al. (2017), the codebook $e \in R^{K \times D}$ has K entries, each with dimensionality D. In our case, D is the same as the embedding dimension of the encoder, resulting in $q(\text{enc}(X))$ with the same shape as $\text{enc}(X)$.

For an input token X_i, its quantized representation is one of the K entries from the codebook $e_{k\in[1,K]}$, where k is determined by a nearest neighbor search in the embedding space, using the encoder output $\text{enc}(X_i)$ as query:

$$k = \arg\min_{j\in[k]} \|\text{enc}(X_i) - e_j\|_2, \quad (1)$$

where $\|\cdot\|_2$ indicates the Euclidean distance.

The quantization step above is vulnerable to index collapse (Kaiser et al., 2018), where only few entries from the embedding table are actively used. On auto-encoding tasks, Kaiser et al. (2018) proposed a countermeasure by breaking down the hidden dimension into multiple slices and quantizing each of them. Specifically, for input token X_i, its encoder hidden state $\text{enc}(X_i)$ is split into S slices:

$$\text{enc}(X_i)_1 \oplus \text{enc}(X_i)_2 \cdots \oplus \text{enc}(X_i)_S, \quad (2)$$

where each slice $\text{enc}(X_i)_{j\in[S]}$ is of D/S dimensions. A nearest neighbor search is conducted for each slice on the corresponding dimensions in the embedding table. The results are then concatenated and form the quantized representation:

$$q(\text{enc}(X_i)_1) \oplus q(\text{enc}(X_i)_2) \cdots \oplus q(\text{enc}(X_i)_S), \quad (3)$$

and passed to the decoder as context. Figure 2 illustrates this process.

The slicing mechanism resembles multi-head attention (Vaswani et al., 2017) in that both split the embedding dimension into subspaces for richer representation. Therefore, we will use the same number of slices as the number of attention heads.

3.2 Soft Discrete Codes

Training Compared to encoder outputs in a continuous space, the quantization module is an *information bottleneck*. In practice, limiting the amount of context information passed to the decoder will likely degrade translation quality. To strike a balance between discretization and performance, we make the discrete codes *soft*, in that the decoder can still access to the richer information prior to quantization by a probability. Specifically, during training, the encoder gives the quantized context $q(\text{enc}(X))$ by probability p, and the raw context $\text{enc}(X)$ by probability $1 - p$. This procedure is illustrated in Figure 1.

In Equation 1, the lookup of index k is a non-differentiable operation. When the encoder passes on the quantized context, in order to train the parameters below the quantization module, we use the straight-through estimator (Bengio et al., 2013) to copy gradients onto the pre-quantization encoder outputs. For the copied gradients to be useful for training, the difference between $\text{enc}(X_i)$ and $q(\text{enc}(X_i))$ should be limited. To achieve this, we use the codebook loss and commitment loss from VQ-VAE (van den Oord et al., 2017):

$$\mathcal{L}_{\text{codebook}} = \|\text{sg}[\text{enc}(X)] - q(\text{enc}(X))\|_2 \quad (4)$$

and

$$\mathcal{L}_{\text{commitment}} = \|\text{enc}(X) - \text{sg}[q(\text{enc}(X))]\|_2, \quad (5)$$

where $\text{sg}[\cdot]$ denotes the stop gradient operation. Intuitively, Equation 4 pushes the codebook entries closer to the points assigned to them, while Equation 5 limits the growth of the encoder hidden states by clipping them to the codebook entries. Each of the terms has weights α_{codebook} and $\alpha_{\text{commitment}}$ to control their importance relative to the main translation objective.

Inference After training with this mechanism, one can expect that the encoder hidden states are well-clustered around a set of codebook entries. At test time, we use the continuous context $\text{enc}(X)$ which still carries more information than the cluster centers represented by the codebook entries. We will verify this property in later experiments (§6).

4 Experimental Setup

To experiment on realistic data volumes, we use the parallel data[1] from the Large-Scale Multilingual Machine Translation Shared Task (Wenzek et al., 2021) from WMT 2021 (Akhbardeh et al., 2021). We focus on small-task-2 on Southeast Asian languages. To study model robustness in zero-shot conditions and the role of language relatedness, we select parallel data between the two high-resource languages: Indonesian (id) and English (en) and three other languages in the Austronesian family: Javanese (jv), Malay (ms), and Filipino/Tagalog (tl). This leads to two data conditions:

- Indonesian-bridge (**ID-BRIDGE**)
- English-bridge (**EN-BRIDGE**)

As pretrained initialization has been shown beneficial in many submissions last year (Yang et al.,

[1] https://data.statmt.org/wmt21/multilingual-task/small_task2_filt_v2.tar.gz

	jv	ms	tl	id	en
jv		340K	662K	644K	2,556K
ms	2M		1,174K	4,060K	12,023K
tl	3M	16M		2,356K	12,348K
en	18M	230M	158M		
id	5M	65M	30M		

Table 2: Number of sentence pairs (above diagonal) and target tokens (below diagonal) from bitext for each languages pair after preprocessing. Data marked with light gray are used in the main experiments.

2021a; Liao et al., 2021; Xie et al., 2021), we initialize the models with the pretrained M2M-124 model provided in the shared task (Wenzek et al., 2021). It is worth noting that M2M-124 has seen parallel data for our *zero-shot* directions, hence zero-shot only describes the condition in our *finetuning* step. This setup is motivated by the observation that existing pretrained models are often trained on massive amounts of data, which are not always feasible to access or store. We therefore treat the pretrained M2M-124 as a given resource, without relying on all its training parallel data. We use this setup to especially study if the models can retain the pretrained knowledge on directions that are zero-shot in finetuning.

4.1 Data

The training parallel data (Wenzek et al., 2021) are compiled from the OPUS platform (Tiedemann, 2012). The specific datasets are listed in Appendix B. As parts of the training data are crawled and therefore rather noisy, we follow the filtering steps opened sourced by Fan et al. (2021), including length filtering, bitext de-duplication, and histogram filtering. An overview of the training data after filtering is in Table 2. Following the evaluation protocol of the shared task (Wenzek et al., 2021), we report spBLEU on the FLoRes-101 (Goyal et al., 2022) devtest set. We additionally report chrF++ (Popović, 2017) as another metric.

4.2 Baselines

Besides comparing to directly training on our baseline model, we also compare to two existing approaches that encourage language-independent representations, both of which have been shown effective in zero-shot translation:

Language-Independent Objective (Pham et al., 2019; Arivazhagan et al., 2019a) applies an additional loss function that enforces the representations for the source and target sentences to be similar. The loss function minimizes the difference between encoded source and target sentences after pooling. Details about the implementation are in Appendix C.1.

Adversarial Language Classifier (Arivazhagan et al., 2019a) aims to remove source language signals from the encoder hidden states, and thereby create more language-independent representations. A language classifier is trained on top of the encoder, and its classification performance is used adversarially on the encoder through a gradient reversal layer (Ganin et al., 2016). Details about the implementation are in Appendix C.2.

4.3 Training and Inference Details

As motivated in §4, we finetune from the small variant of M2M-124 with 175M parameters. This model has a vocabulary size of 256K, 6 layers in both the encoder and decoder, 16 attention heads, embedding dimension of 512 and inner dimension of 2048. As the training data for different languages are very unbalanced, we use temperature-based sampling (Arivazhagan et al., 2019b) with coefficient 5.0, which heavily upsamples low-resource directions and is recommended for unbalanced data conditions (Arivazhagan et al., 2019b; Tang et al., 2021). Additional details are in Appendix A.

For our codebook approach, we use 10K codebook entries. Initial trials with a size of 1K gave worse performance, while 40K heavily reduced training speed. We choose 16 slices[2] for the codebook, the same value as the number of attention heads. We keep these two values identical as both slicing and multi-head attention breaks the embedding dimension into multiple subspaces of lower dimensionality. The scale on the codebook loss and commitment loss (α_{codebook} and $\alpha_{\text{commitment}}$) are 1.0 and 1.001. We found the model sensitive to increasing $\alpha_{\text{commitment}}$, where higher values leads to index collapse[3]. After exponentially decreasing it to approach 1.0, we settled at 1.001. For the probability of seeing the continuous encoder context, with a search among {0.1, 0.5, 0.7 0.9}, we found 0.9 and 0.5 the best parameters for ID-BRIDGE and EN-BRIDGE respectively.

We implement our approach and the two baselines (§4.2) with FAIRSEQ (Ott et al., 2019)[4].

[2] Initial experiments on smaller datasets showed weaker translation performance with 2 and 4 slices.

[3] A potential reason is the encoder parameters are updated too aggressively by the commitment loss in these cases.

[4] Code available at: https://github.com/dannigt/

ID	Model	Avg. spBLEU(↑) (left) and chrF++(↑) (right)							
		{jv, ms, tl} →X		X→ {jv, ms, tl}		Y↔Z		Avg. (all dir.)	
	Id-Bridge (X=id)								
(1)	random initialization	27.5	52.7	24.2	49.4	15.8	41.5	20.8	46.3
(2)	`M2M-124` (Fan et al., 2021; Goyal et al., 2022)	20.0	45.7	14.7	38.9	9.9	34.3	13.6	38.3
(3)	↪ parallel data (no data for **Y↔Z**)	27.1	52.5	24.2	49.6	17.7	43.3	21.7	47.2
(3.1)	+ language-independent objective	27.1	52.4	24.2	49.6	18.4	43.8	22.0	47.4
(3.2)	+ adversarial language classifier	27.5	52.9	24.1	49.6	18.4	44.2	22.1	47.7
(3.3)	+ codebook (ours)	27.2	52.4	23.6	49.2	18.3	44.0	21.9	47.4
	En-Bridge (X=en)								
(4)	random initialization	27.0	51.1	27.8	51.6	6.8	24.5	17.1	37.9
(5)	`M2M-124` (Fan et al., 2021; Goyal et al., 2022)	19.6	43.6	14.0	37.5	9.9	34.3	13.3	37.4
(6)	↪ parallel data (no data for **Y↔Z**)	28.1	51.8	27.6	51.8	5.1	20.3	16.5	36.1
(6.1)	+ language-independent objective	27.9	51.7	27.2	51.4	17.3	42.8	22.4	47.2
(6.2)	+ adversarial language classifier	27.6	51.5	27.1	51.5	17.2	42.8	22.3	47.2
(6.3)	+ codebook (ours)	26.8	50.6	26.3	50.9	15.2	39.3	20.9	45.0

Table 3: Translation quality in spBLEU(↑) and chrF++(↑). "↪" indicates finetuning on the parallel data (Id-Bridge or En-Bridge; §4). Pivoting through the bridge language for $Y \leftrightarrow Z$ directions scores 19.7, 17.5 spBLEU and 44.9, 42.8 chrF++ for Id-Bridge and En-Bridge respectively using the systems in rows (1) and (4).

5 Main Results

We first discuss the translation performance of our multilingual systems (§5.1), and then use the learned discrete codes to investigate cross-lingual knowledge-sharing of the trained models (§5.2).

5.1 Translation Performance

Baseline Conditions To set the upcoming results in context, we first present the performance of training without additional improvements in rows (1)-(3) and (4)-(6) of Table 3. Rows (1) and (4) show the performance of training with random initialization. This corresponds to a condition where we have parallel data but no pretrained resources. On the other side of the spectrum, in row (2) and (5), we report the results of directly running inference on the pretrained `M2M-124` model. This corresponds to another extreme where we have access to pretrained models but cannot additionally train on parallel data. In rows (3) and (6), we combine the best of two worlds: initializing with pretrained model and finetuning on parallel data. For supervised directions, pretraining mainly improves →English directions: In the En-Bridge condition, initializing with `M2M-124` gains 1.1 spBLEU over random initialization, from 27.0 to 28.1 spBLEU. For other supervised directions, however, we do not observe gains from pretraining. This could be related to the pretrained model being particularly strong at decoding English. For zero-shot directions in our setup (these directions are seen in training by the pretrained model), as they are comparatively low-resource among all the directions covered in `M2M-124`, out-of-box translation quality on these directions is relatively low, with an average of 9.9 spBLEU. However, when finetuning, we see a striking difference between Id-Bridge and En-Bridge: there is a large gain from 9.9 to 17.7 spBLEU with the former, but a degradation from 9.9 to 5.1 spBLEU for the latter. We study this phenomenon next.

Impact of Bridge Languages For En-Bridge, the finetuning step causes catastrophic forgetting of the zero-shot directions (−4.8 spBLEU). On the other hand, for the Id-Bridge condition, pure finetuning leads to substantial improvements in *both* supervised and zero-shot directions. The gain from 9.9 to 17.7 spBLEU in the $Y \leftrightarrow Z$ directions is particularly noteworthy since the model has not seen parallel data for these directions in finetuning. This indicates that the growth in supervised directions brings zero-shot directions forward too. Moreover, on these directions, pretraining also gives large gain of 1.9 spBLEU over random initialization. Overall, the observations suggest that incorporating a similar language as bridge is beneficial to re-using pretrained knowledge. Furthermore, given that the amount of parallel data in the En-Bridge condition is nearly 4 times of that in the Id-Bridge condition, using a similar bridge language also appears to be more *data-efficient*. This likely related to all translation directions being similar, therefore easing the multilingual learning task.

`fairseq/tree/master/examples/quant`

Impact of Using Codebooks Compared to pure finetuning in rows (3) and (6), by incorporating the codebook we improve zero-shot translation by 0.6 and 10.1 spBLEU for ID-BRIDGE andEN-BRIDGE respectively. Compared to the two existing approaches, namely language-independent objective and adversarial language classifier in rows $(*.1)$ and $(*.2)$, our approach performs on par with them for ID-BRIDGE, achieving 18.3 spBLEU for $Y \leftrightarrow Z$ directions and 21.9 spBLEU over all directions. In the more challenging EN-BRIDGE condition, we fall behind the two other approaches by around 2.0 spBLEU on zero-shot directions. Using a language identifier[5] (Costa-jussà et al., 2022), we found that the culprit here is still off-target translation, where some test sentences were translated to an incorrect language. While our codebook approach reduces the proportion of off-target sentences from 87.4% to 13.1% compared to the pure finetuning baseline in row (6), the figure is still higher than the 4.7% achieved by the two alternative models in rows (6.1) and (6.2). Despite this gap, an advantage of our approach is easier analyses of learned representations, which we will now leverage to investigate why the two data conditions come with very distinct zero-shot behavior.

5.2 Using Discrete Codes to Interpret Learned Representations

Since our codebook approach allows easier interpretation of model hidden representations, we take advantage of this characteristic to answer the following question: *why is the* ID-BRIDGE *data condition more performant despite using less data?*

Formalization To this end, we first extract the discrete codes for all source languages on the test set[6]. Given a total of S slices, a sentence with t tokens $X_{1,\dots,t}$ is represented as S sets of discrete tokens $T^s_{1,\dots,t}$ for slice s, where $s \in [S]$. Between two sets of semantically identical sentences (e.g. multiway test sets in two different languages), we can compare the discrete codes by examining: 1) their overlap and 2) the difficulty of transforming one set to another. The results quantify the similarity between the two sets of codes, and hence the model representations for the two source languages.

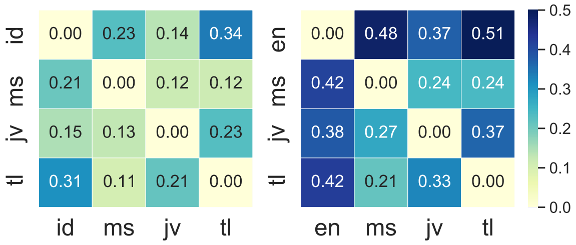


Figure 3: KL divergence(↓) of code distribution for the ID-BRIDGE (left) and EN-BRIDGE (right) setup. *Lower* values indicate a higher degree of sharing. ID-BRIDGE results in more sharing not only between itself and {ms, jv, tl} but also among {ms, jv, tl}.

Discrete Code Distribution For each slice, we normalize the code occurrences into a probability distribution. The distribution P is defined by:

$$p(c_i) = \frac{\text{frequency}(c_i)}{\sum_{c_j \in [C]} \text{frequency}(c_j)}, \qquad (6)$$

where c_i is a discrete code from the set $[C]$. For a pair of languages i and j, we then compute the KL divergence between their code distributions P_i and P_j:

$$D^{(i,j)}_{\text{KL}} = (P_i || P_j). \qquad (7)$$

Figure 3 depicts the KL divergence of code distribution averaged over all slices. A comparison of the En- and ID-BRIDGE setup exhibits several major differences. First, the clearly prominent first row and column in EN-BRIDGE shows that its bridge-language is represented very differently from all other languages ({ms, jv, tl}). For the ID-BRIDGE counterpart, the difference between the bridge language and the remaining languages is much milder. Second, but perhaps more importantly, among the languages used in zero-shot directions ({ms, jv, tl}), the amount of sharing is also higher under the ID-BRIDGE setup. This finding is crucial as the raw tokens for {ms, jv, tl} are identical between the ID-BRIDGE and EN-BRIDGE setup. Therefore, the higher degree of sharing is clearly an outcome of the model creating its representations differently. Overall, these results show that the choice of the bridge language not only impacts the knowledge-sharing mechanism between itself and the remaining languages, but also for the remaining languages in the model.

Discrete Code Translation The code distribution analysis above makes a simplified assumption by considering the discrete codes as a *bag of words*.

[5] https://github.com/facebookresearch/fairseq/tree/nllb#lid-model

[6] The FLoRes-101 test set is multiway. Therefore the semantic meanings of the sentences are the same.

To additionally assess the *structural (dis)similarity* between the code representations for different languages, we consider the task of *translating* the discrete codes of a language to another.

While a constructed language like Interlingua would create the same representations for the source sentences with identical meanings, our discrete code representation is not yet invariant to the source language. Nevertheless, we do expect them to be more abstracted from the source sentences, making the translation task easier than directly between the raw tokens. Here we train a translation model on the discrete codes and use the test performance to quantify how similarly the source languages are represented. When the representations are more different from each other, i.e. language-specific, the translation quality on the discrete is expected to be lower.

Specifically, we randomly sample 100K sentence pairs[7] for each translation direction in the experiments of Table 3 extract their discrete codes assigned by the trained models (rows (3.3) and (6.3) of Table 3), and train a new Transformer-base (Vaswani et al., 2017) to translate between the extracted codes of different languages. We flatten the slices, therefore making each source token represented by 16 discrete codes. After training for 200K steps, we report BLEU scores on the test set, which is also converted to discrete codes. The results are shown in Figure 4. First, the translation task is clearly easier on the discrete codes derived from the ID-BRIDGE system. Second, the scores differences are especially prominent when translating out of Malay (ms) and Javanese (jv), which are more related to Indonesian than Filipino/Tagalog (tl). Along with the results from the code overlap, our results show that using a similar bridge language results in higher knowledge-sharing not only syntactically but also structurally, especially between related languages.

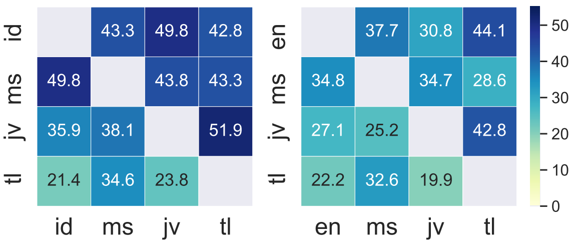
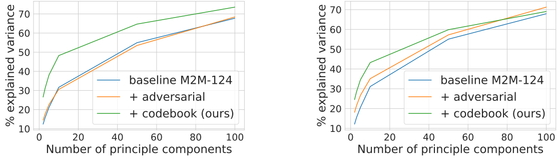


Figure 4: BLEU(↑) scores of translating between discrete codes for the ID-BRIDGE (left) and EN-BRIDGE (right) setup. *Higher* values indicate a higher degree of sharing. In general it is easier to translate the codes for the ID-BRIDGE setup, indicating more structural similarity between the representations.



(a) ID-BRIDGE (b) EN-BRIDGE

Figure 5: Our codebook approach creates better-clustered encoder hidden states, as shown by a much higher percentage of variance explained by PCA compared to both the baseline and a strong alternative approach (adversarial language classifier).

6 Analyses on Learned Discrete Codes

Next we further investigate the discrete codes regarding its usefulness for the learned representations (§6.1) as well as the translation task (§6.2).

6.1 How well-clustered are the hidden states?

As motivated in §3, although at inference time we use the continuous encoder hidden states instead of the cluster centers, the soft discrete codes will still enforce encoder hidden states into clusters, thereby resembling a discrete structure. To verify whether the encoder latent space indeed becomes more discretized with our approach, we analyze the encoder hidden states on the test set using Principle Component Analysis (PCA). If the data points representing the encoder outputs are well-clustered, a larger percentage of their variance should be explained by the learned principle components. As shown in Figure 5, our approach (marked with green line) consistently leads to higher proportions of explained variances compared to the baseline `M2M-124`, as well as the strong alternative approach with the adversarial language classifier. These results therefore confirm the effectiveness of our soft discrete code approach in enforcing discrete structures in the encoder latent space.

6.2 Meaningfulness of Clusters Centers

Recall that at inference time our soft discrete code model uses the encoder hidden states prior to discretiztaion, although it does use both pre- and post-discretization encoder context in training. A main reason of doing so is that discretizing the encoder

[7] The training data (Table 2) allow us to use 340K sentences. We sampled 100K for faster experiment iteration.

hidden states to cluster centers creates an information bottleneck that limits model expressiveness. Despite the expected performance degradation, we are nonelessness interested in *quantifying* how much information is lost by using the cluster centers as context instead. In other words, the question is *how meaningful are the cluster centers for the translation task?* In Table 4, we report the results of using the cluster centers as context for the decoder at inference time. Compared to using the encoder hidden states, we see a degradation of 4.1 and 1.7 and spBLEU for ID-BRIDGE and EN-BRIDGE respectively. This indicates that the cluster centers are still relevant for the translation task, although much less powerful than the encoder hidden states prior to discretization. It also rules out the possibility of the learned codes being trivial repetitions, which would otherwise have been detrimental to the translation performance.

Encoder States at Inference	**Avg. spBLEU(↑)**			
	→X	**X→**	**Y↔Z**	**Avg.**
ID-BRIDGE (X=id)				
encoder states (Tab. 3 row (3.3))	27.2	23.6	18.3	21.9
cluster centers	22.8	20.0	14.3	17.8
EN-BRIDGE (X=en)				
encoder states (Tab. 3 row (6.3))	26.8	26.3	15.2	20.9
cluster centers	24.3	24.6	13.9	19.2

Table 4: At inference time, using cluster centers instead of the clustered encoder states degrades performance by 1.7-4.1 spBLEU. Despite the degradation, the scores show that translation from the clusters centers is still meaningful. This also rules out the possibility of the learned codes collapsing to trivial repetitions.

7 Analyses on Zero-Shot Translation

Our experiments so far use single-bridge languages and are evaluated in part on zero-shot directions. We now study the impact when either of the two conditions changes: 1) when parallel data is available for previously zero-shot directions; 2) when using multiple bridge languages.

7.1 When does zero-shot translation match the performance on parallel data?

Zero-shot conditions could be avoided by creating synthetic data from back-translation (Sennrich et al., 2016; Zhang et al., 2020) or mining additional parallel data (Fan et al., 2021; Freitag and Firat, 2020). Both approaches introduce additional workflows into the pipeline of building translation systems. We are therefore interested in the following question: *How much parallel data do we need to perform better than direct zero-shot translation?*

The training corpora from the shared task (§4.1) provides an oracle condition to answer this question. As shown in Table 2, the oracle parallel data amounts to 2.2M sentences in total (340K for jv-ms, 662K for jv-tl, and 1.2M sentences for ms-tl). We take 100%, 10% and 1% of the oracle parallel data and training systems together with the original data and train multilingual systems with the same configuration as rows (3) and (6) of Table 3. The results are shown in Table 5.

To our surprise, adding 1% oracle bitext (22K sentence pairs in total) of the previously zero-shot directions already results in comparable performance to the best zero-shot performance (18.4 and 17.3 spBLEU for ID-BRIDGE and EN-BRIDGE respectively). However, this comes with some degradation on supervised directions of 0.4 spBLEU for ID-BRIDGE and 0.7 for EN-BRIDGE. This is likely due to the temperature-based sampling aggressively upsampling the extremely low-resource directions, meanwhile causing the model to deprioritize other higher-resource directions. When increasing oracle bitext to 10% (220K sentence pairs in total), the system outperforms direct zero-shot performance. Lastly, the additional gain appear to diminish when going from 10% to all oracle data. For ID-BRIDGE, the performance appears saturated at 10%: adding the remaining 90% parallel data does not give additional gain. On the contrary, For EN-BRIDGE, the system appears to still improve, especially on $Y \leftrightarrow Z$ directions (+0.5 spBLEU). The performance on these directions nevertheless still falls behind the ID-BRIDGE direction by 0.8 spBLEU (18.9 vs 19.7 spBLEU). An explanation is that the EN-BRIDGE system requires more data to train as a result of the bridge language being very distant to the rest, thereby increasing the difficulty of multitasking over all the translation directions. This echos with the previous finding that using related bridge languages eases the multilingual translation task and increases knowledge-sharing (§5.1).

7.2 Do multiple bridge languages bring additional gains?

While the experiments so far are based on single bridge languages, in practice we often have access to multi-bridge parallel data. Indeed, recent

Oracle Bitext	**Avg. spBLEU(↑)**			
	→X	**X→**	**Y↔Z**	**Avg.**
Id-Bridge (X=id)				
best zero-shot (Tab. 3 row (3.2))	27.5	24.1	18.4	22.1
1%	26.9	23.9	18.4	21.9
10%	27.3	24.5	19.7	22.8
100% (2.2M bitext)	26.6	24.8	19.7	22.7
En-Bridge (X=en)				
best zero-shot (Tab. 3 row (6.1))	27.9	27.2	17.3	22.4
1%	27.0	26.8	17.1	22.0
10%	27.5	27.4	18.4	22.9
100% (2.2M bitext)	27.7	27.5	18.9	23.2

Table 5: Impact of adding oracle parallel data for the previously zero-shot directions. Adding 10% parallel data (roughly 220K sentence pairs in our case) surpasses the best performance on direct zero-shot translation.

Data Condition		**Avg. spBLEU(↑)**			
		→X	**X→**	**Y↔Z**	**Avg.**
Multi-Bridge	X= id	27.0	24.3	18.3	21.9
	X= en	27.8	27.7		23.0
Only Id-Bridge (Tab. 3 row (3))		27.1	24.2	17.7	21.7
Only En-Bridge (Tab. 3 row (6))		28.1	27.6	5.1	16.5

Table 6: Results of using multiple bridges (combining Id-Bridge and En-Bridge). Despite substantial gains over En-Bridge, the multi-bridge system only gives a mild improvement in zero-shot performance ($Y \leftrightarrow Z$) over the Id-Bridge system.

works (Freitag and Firat, 2020; Fan et al., 2021) have shown success on large-scale fully-connected models, as well as evidence of multi-bridge outperforming the English-bridge condition (Rios et al., 2020). What remains unclear is whether there is a synergy when combining the parallel data from several single-bridge conditions. We investigate this hypothesis by training a multi-bridge system, combing the data from our Id-Bridge and En-Bridge setup. As shown in Table 6, for supervised directions of $\rightarrow X$ and $X \rightarrow$, there is no clear difference between the performance of the multi-bridge system and that of the single-bridge ones. For zero-shot directions ($Y \leftrightarrow Z$), while multi-bridge gains substantially over En-Bridge (18.3 from 5.1 spBLEU), there is only a slight gain over Id-Bridge. Given that the multi-bridge model more than doubles the training time of Id-Bridge, the little performance difference to the multi-bridge system shows that choosing a bridge language related to the remaining languages is a data-efficient way to achieve strong zero-shot performance.

8 Conclusion

In this work, we focus on learning to represent source sentences of multilingual NMT models by discrete codes. On multiple large-scale experiments, we show that our approach not only increase the model robustness in zero-shot conditions, but also offers more interpretable intermediate representations. We leverage the latter property to investigate the role of bridge languages, and show that using a more related bridge language leads to increased knowledge-sharing, not only between the bridge language and remaining but also between all other languages involved in training.

A limitation is that the discrete codes only give a mechanism to compare hidden representations, but are not directly interpretable by humans. A potential improvement would be to use an existing codebook that corresponds to an actual human language. Besides this, as next steps, we plan to improve the generation process of the discrete codes. The first direction is to make the code lookup conditionally-dependent along the time dimension and learn to shrink the sequence length of the discrete codes, thereby creating a more compact representation. Another direction is to explicitly incentivize more shared codes between different, and especially related, languages during training. This would bring the discrete codes closer to a language-independent representation.

Acknowledgement We thank James Cross and Paco Guzmán for their feedback in the early stage of this work. We thank the anonymous reviewers for their detailed and insightful feedback. We also thank Ngoc Quan Pham, Tu Anh Dinh, and Sai Koneru for feedback on the paper draft.

References

Ahmed Abdelali, Francisco Guzman, Hassan Sajjad, and Stephan Vogel. 2014. The AMARA corpus: Building parallel language resources for the educational domain. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1856–1862, Reykjavik, Iceland. European Language Resources Association (ELRA).

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. In neural machine translation, what does transfer learning transfer? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7701–7710, Online. Association for Computational Linguistics.

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Maruan Al-Shedivat and Ankur Parikh. 2019. Consistency by agreement in zero-shot neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1184–1197, Minneapolis, Minnesota. Association for Computational Linguistics.

Antonios Anastasopoulos, Ondřej Bojar, Jacob Bremerman, Roldano Cattoni, Maha Elbayad, Marcello Federico, Xutai Ma, Satoshi Nakamura, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Alexander Waibel, Changhan Wang, and Matthew Wiesner. 2021. FINDINGS OF THE IWSLT 2021 EVALUATION CAMPAIGN. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 1–29, Bangkok, Thailand (online). Association for Computational Linguistics.

Antonios Anastasopoulos, Alessandro Cattelan, Zi-Yi Dou, Marcello Federico, Christian Federmann, Dmitriy Genzel, Franscisco Guzmán, Junjie Hu, Macduff Hughes, Philipp Koehn, Rosie Lazar, Will Lewis, Graham Neubig, Mengmeng Niu, Alp Öktem, Eric Paquin, Grace Tang, and Sylwia Tur. 2020. TICO-19: the translation initiative for COvid-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, Online. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Roee Aharoni, Melvin Johnson, and Wolfgang Macherey. 2019a. The missing ingredient in zero-shot neural machine translation. *CoRR*, abs/1903.07091.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019b. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432.

Ting-Rui Chiang, Yi-Pei Chen, Yi-Ting Yeh, and Graham Neubig. 2022. Breaking down multilingual machine translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2766–2780, Dublin, Ireland. Association for Computational Linguistics.

Christos Christodoulopoulos and Mark Steedman. 2015. A massively parallel corpus: the bible in 100 languages. *Lang. Resour. Evaluation*, 49(2):375–395.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *CoRR*, abs/2207.04672.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational*

Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5960–5969, Online. Association for Computational Linguistics.

Muhammad N. ElNokrashy, Amr Hendy, Mohamed Maher, Mohamed Afify, and Hany Hassan Awadalla. 2022. Language tokens: A frustratingly simple approach improves zero-shot performance of multilingual translation. *CoRR*, abs/2208.05852.

Carlos Escolano, Marta R. Costa-Jussà, and José A. R. Fonollosa. 2021. From bilingual to multilingual neural-based machine translation by incremental training. *Journal of the Association for Information Science and Technology*, 72(2):190–203.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *The Journal of Machine Learning Research*, 22:107:1–107:48.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Markus Freitag and Orhan Firat. 2020. Complete multilingual neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 550–560, Online. Association for Computational Linguistics.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O.K. Li. 2019. Improved zero-shot neural machine translation via ignoring spurious correlations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1258–1268, Florence, Italy. Association for Computational Linguistics.

Thanh-Le Ha, Jan Niehues, and Alex Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. In *Proceedings of the 13th International Conference on Spoken Language Translation*, Seattle, Washington D.C. International Workshop on Spoken Language Translation.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Lukasz Kaiser, Samy Bengio, Aurko Roy, Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, and Noam Shazeer. 2018. Fast decoding in sequence models using discrete latent variables. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2395–2404. PMLR.

Zae Myung Kim, Laurent Besacier, Vassilina Nikoulina, and Didier Schwab. 2021. Do multilingual neural machine translation models contain language pair specific attention heads? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2832–2841, Online. Association for Computational Linguistics.

Rebecca Knowles and Samuel Larkin. 2021. NRC-CNRC systems for Upper Sorbian-German and Lower Sorbian-German machine translation 2021. In *Proceedings of the Sixth Conference on Machine Translation*, pages 999–1008, Online. Association for Computational Linguistics.

Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual NMT representations at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1565–1575, Hong Kong, China. Association for Computational Linguistics.

Baohao Liao, Shahram Khadivi, and Sanjika Hewavitharana. 2021. Back-translation for large-scale multilingual machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 418–424, Online. Association for Computational Linguistics.

Jindřich Libovický and Alexander Fraser. 2021a. Findings of the WMT 2021 shared tasks in unsupervised

MT and very low resource supervised MT. In *Proceedings of the Sixth Conference on Machine Translation*, pages 726–732, Online. Association for Computational Linguistics.

Jindřich Libovický and Alexander Fraser. 2021b. The LMU Munich systems for the WMT21 unsupervised and very low-resource translation task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 989–994, Online. Association for Computational Linguistics.

Danni Liu and Jan Niehues. 2021. Maastricht university's multilingual speech translation system for IWSLT 2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 138–143, Bangkok, Thailand (online). Association for Computational Linguistics.

Danni Liu, Jan Niehues, James Cross, Francisco Guzmán, and Xian Li. 2021a. Improving zero-shot translation by disentangling positional information. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1259–1273, Online. Association for Computational Linguistics.

Hui Liu, Danqing Zhang, Bing Yin, and Xiaodan Zhu. 2021b. Improving pretrained models for zero-shot multi-label text classification through reinforced label hierarchy reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1051–1062, Online. Association for Computational Linguistics.

Aaron Mueller, Garrett Nicolai, Arya D. McCarthy, Dylan Lewis, Winston Wu, and David Yarowsky. 2020. An analysis of massively multilingual neural machine translation for low-resource languages. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3710–3718, Marseille, France. European Language Resources Association.

Mohammad Norouzi, Tomás Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg Corrado, and Jeffrey Dean. 2014. Zero-shot learning by convex combination of semantic embeddings. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Ngoc-Quan Pham, Tuan Nam Nguyen, Thanh-Le Ha, Sebastian Stüker, Alexander Waibel, and Dan He. 2021. Multilingual speech translation KIT @ IWSLT2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 154–159, Bangkok, Thailand (online). Association for Computational Linguistics.

Ngoc-Quan Pham, Jan Niehues, Thanh-Le Ha, and Alexander Waibel. 2019. Improving zero-shot translation with language-independent constraints. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 13–23, Florence, Italy. Association for Computational Linguistics.

Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual adapters for zero-shot neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4465–4470, Online. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Alessandro Raganato, Raúl Vázquez, Mathias Creutz, and Jörg Tiedemann. 2021. An empirical investigation of word alignment supervision for zero-shot multilingual neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8449–8456, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Annette Rios, Mathias Müller, and Rico Sennrich. 2020. Subword segmentation and a single bridge language affect zero-shot neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 528–537, Online. Association for Computational Linguistics.

Bernardino Romera-Paredes and Philip H. S. Torr. 2015. An embarrassingly simple approach to zero-shot learning. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2152–2161. JMLR.org.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. 2013. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 935–943.

Bokyung Son and Sungwon Lyu. 2020. Sparse and decorrelated representations for stable zero-shot NMT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2260–2266, Online. Association for Computational Linguistics.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. 2020. Transformer VQ-VAE for unsupervised unit discovery and speech synthesis: Zerospeech 2020 challenge. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 4851–4855. ISCA.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6306–6315.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Guillaume Wenzek, Vishrav Chaudhary, Angela Fan, Sahir Gomez, Naman Goyal, Somya Jain, Douwe Kiela, Tristan Thrush, and Francisco Guzmán. 2021. Findings of the WMT 2021 shared task on large-scale multilingual machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 89–99, Online. Association for Computational Linguistics.

Liwei Wu, Shanbo Cheng, Mingxuan Wang, and Lei Li. 2021. Language tags matter for zero-shot neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3001–3007, Online. Association for Computational Linguistics.

Yongqin Xian, Bernt Schiele, and Zeynep Akata. 2017. Zero-shot learning - the good, the bad and the ugly. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3077–3086. IEEE Computer Society.

Wanying Xie, Bojie Hu, Han Yang, Dong Yu, and Qi Ju. 2021. TenTrans large-scale multilingual machine translation system for WMT21. In *Proceedings of the Sixth Conference on Machine Translation*, pages 439–445, Online. Association for Computational Linguistics.

Jian Yang, Shuming Ma, Haoyang Huang, Dongdong Zhang, Li Dong, Shaohan Huang, Alexandre Muzio, Saksham Singhal, Hany Hassan, Xia Song, and Furu Wei. 2021a. Multilingual machine translation systems from Microsoft for WMT21 shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 446–455, Online. Association for Computational Linguistics.

Yilin Yang, Akiko Eriguchi, Alexandre Muzio, Prasad Tadepalli, Stefan Lee, and Hany Hassan. 2021b. Improving multilingual translation by representation and gradient regularization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7266–7279, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021. Share or not? learning to schedule language-specific capacity for multilingual translation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Biao Zhang and Rico Sennrich. 2021. Edinburgh's end-to-end multilingual speech translation system for IWSLT 2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 160–168, Bangkok, Thailand (online). Association for Computational Linguistics.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1628–1639, Online. Association for Computational Linguistics.

Changfeng Zhu, Heng Yu, Shanbo Cheng, and Weihua Luo. 2020. Language-aware interlingua for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1650–1655, Online. Association for Computational Linguistics.

A Additional Training and Inference Details

When training, one optimization step happens after 16384 tokens. We use the Adam optimizer with betas $(0.9, 0.98)$. The learning rate is 0.0001 with the inverse squared root schedule and 2500 warmup steps. As for regularization parameters, we use label smoothing of 0.1, dropout of 0.3, and attention dropout 0.1. The models are trained for 500K updates in total. An exception is the MULTIBRIDGE experiment with more training data, where we trained for 800K updates in total. For inference, we decode with a beam size of 5.

B Dataset Details

The training parallel data include the following corpora: bible-uedin (Christodoulopoulos and Steedman, 2015), (Multi)CCAligned (El-Kishky et al., 2020), Gnome[8], ELRC[9], KDE4[10], GlobalVoices[11], OpenSubtitles[12], QED (Abdelali et al., 2014), MultiParaCrawl[13], TED2020[14], Tanzil[15], Tatoeba[16], Ubuntu[17], WikiMatrix (Schwenk et al., 2021), wikimedia[18], and TICO-19 (Anastasopoulos et al., 2020).

C Implementation of Baselines

C.1 Language-Independent Objective

We chose meanpool and L2 distance for the similarity loss since it gave better or more consistent performance in initial experiments. As for the weight of the language-independent objective, we used 1.0 following Pham et al. (2019).

C.2 Adversarial Classifier

We extend the adversarial language classification approach from Arivazhagan et al. (2019a) for robust training. Specifically, we use a modified loss when adversarially training the encoder. Moreover, we apply the language classification on the token level to remove the need for selecting a pooling method. The classifier minimizes the cross-entropy loss when predicting the language labels:

$$\mathcal{L}_{\text{classifier}} = -\sum_{c=1}^{L} y_c \text{log}(p_c), \tag{8}$$

where L is the number of classes to predict, y_c is a binary indicator whether the true language label is c, and p_c is the predicted probability for the instance belonging to language c.

Removing source language signals from the encoder representations can be achieved by a gradient reversal layer (Ganin et al., 2016) from the language classification. An issue with the standard classification loss in Equation 8 is that, when the classifier is performing well, the loss landscape is rather flat, causing minimal gradient flow to the encoder. In fact, when the classifier predicts the source languages accurately, we instead need large gradients to update the encoder representations as they contain high amounts of language signals. Therefore, when updating the encoder parameters adversarially, we use the modified loss:

$$\mathcal{L}_{\text{adv\_classifier}} = \sum_{c=1}^{L} y_c \text{log}(1 - p_c), \tag{9}$$

which in effect mirrors Equation 8 by the horizontal axis and the vertical line defined by $x = 0.5$. With the modified loss, the optimization direction does not change, but the gradient is larger when the classifier is performing well.

The translation model is then trained with:

$$\mathcal{L}_{\text{encoder\_decoder}} = \mathcal{L}_{\text{MT}} + \mathcal{L}_{\text{adv\_classifier}}. \tag{10}$$

For training stability, we alternate the optimization of the classifier (Equation 8) and the main encoder-decoder parameters (Equation 10). Optimizing them jointly would otherwise lead to co-adaptation of the parameters of the translation and classification module and empirically causes training instability.

[8] https://opus.nlpl.eu/GNOME.php

[9] https://opus.nlpl.eu/ELRC.php

[10] https://opus.nlpl.eu/KDE4.php

[11] https://opus.nlpl.eu/GlobalVoices.php

[12] https://opus.nlpl.eu/OpenSubtitles-v2018.php

[13] https://opus.nlpl.eu/MultiParaCrawl.php

[14] https://opus.nlpl.eu/TED2020.php

[15] https://opus.nlpl.eu/Tanzil.php

[16] https://opus.nlpl.eu/Tatoeba.php

[17] https://opus.nlpl.eu/Ubuntu.php

[18] https://opus.nlpl.eu/wikimedia.php

Don't Discard Fixed-Window Audio Segmentation in Speech-to-Text Translation

Chantal Amrhein[1] and **Barry Haddow**[2]
[1]Department of Computational Linguistics, University of Zurich
[2]School of Informatics, University of Edinburgh
amrhein@cl.uzh.ch, bhaddow@ed.ac.uk

Abstract

For real-life applications, it is crucial that end-to-end spoken language translation models perform well on continuous audio, without relying on human-supplied segmentation. For *online* spoken language translation, where models need to start translating before the full utterance is spoken, most previous work has ignored the segmentation problem. In this paper, we compare various methods for improving models' robustness towards segmentation errors and different segmentation strategies in both offline and online settings and report results on translation quality, flicker and delay. Our findings on five different language pairs show that a simple fixed-window audio segmentation can perform surprisingly well given the right conditions.[1]

1 Introduction

End-to-end spoken language translation (SLT) has seen considerable advances in recent years. To apply these findings to real online and offline SLT settings, we need to be able to process continuous audio input. However, most previous work on end-to-end SLT makes use of human-annotated, sentence-like gold segments both at training and test time which are not available in real-life settings. Unfortunately, SLT models that were trained on such gold segments often suffer a noticeable quality loss when applied to artificially split audio segments (Zhang et al., 2021; Tsiamas et al., 2022b). This also highlights that a good segmentation is more important for SLT than for automatic speech recognition (ASR) because we need to split the audio into "translatable units". For a cascade system, a segmenter/punctuator can be inserted between the ASR and machine translation (MT) model (Cho et al., 2017) in order to create suitable segments for the MT model. However for end-to-end SLT systems, it is still not clear how to best translate continuous input.

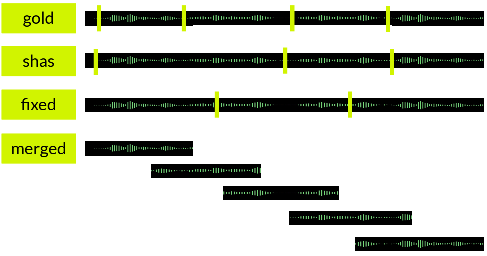


Figure 1: Visualisation of the different audio segmentation methods studied in this paper.

Solving this problem is very much an active research field that has mainly been tackled from two sides: (1) improving SLT models to be more robust towards segmentation errors (Gaido et al., 2020; Li et al., 2021; Zhang et al., 2021) and (2) developing strategies to split streaming audio into segments that resemble the training data more closely (Gaido et al., 2021; Tsiamas et al., 2022b). Both types of approaches were successfully used in recent years for the IWSLT offline SLT shared task (Ansari et al., 2020; Anastasopoulos et al., 2021, 2022) to translate audio without gold segmentations. However, they have not yet been tested systematically in the online SLT setup where translation starts before the full utterance is spoken. Recent editions of the IWSLT simultaneous speech translation shared task focused more on evaluation using the gold segmentation rather than unsegmented audio (Anastasopoulos et al., 2021, 2022). Segmenting streaming audio is especially interesting in online SLT because aside from effects on translation quality, different segmentations can also influence the delay (or latency) of the generated translation.

In this paper, we aim to fill this gap and focus on the end-to-end online SLT setup. We suspect that there is an interplay between more robust models and better segmentation strategies

[1] We publicly release our code and model outputs here: https://github.com/ZurichNLP/window_audio_segmentation

and that an isolated comparison may not be informative enough. Consequently, we explore different combinations of these two approaches for two different SLT models and present results in five language pairs. Figure 1 shows the four segmentation methods we study in this work (see also Section 3.3). Our experiments follow the popular retranslation approach (Niehues et al., 2016, 2018; Arivazhagan et al., 2020a,b) where a partial segment is retranslated every time new audio becomes available. Retranslation has the advantage of being a simple approach to online SLT, which can use a standard MT inference engine. As a side-effect, the previous translation can change in later retranslations and the resulting "flicker" (i.e. sudden translation changes in the output of previous time steps) is also considered in our evaluation of different strategies.

Our main contributions are:

- We explore various combinations of segmentation strategies and robustness-finetuning approaches for translating unsegmented audio in an online SLT setup.
- We find that the advantage of dedicated audio segmentation models over a fixed-window approach becomes much smaller if the translation model is context-aware, and merging translations of overlapping windows can perform comparatively to the gold segmentation.
- We discuss issues with the evaluation of delay in an existing evaluation toolkit for retranslation when different segmentations are used and show how these can be mitigated.

2 Related Work

In recent years, the IWSLT shared task organisers have stopped providing gold segmented test sets for the offline speech translation task which has lead to increased research focus on audio segmentation (Ansari et al., 2020; Anastasopoulos et al., 2021, 2022). One obvious strategy to segment audio is to create fixed windows of the same duration, but previous research has mostly relied on more elaborate methods. Typically, methods with voice activity detection (VAD) (Sohn et al., 1999) were employed to identify natural breaks in the speech signal. However, VAD models do not guarantee breaks that align with complete utterances and can produce segments that are too long or too short which is why hybrid approaches that also consider the length of the predicted utterance can be helpful (Potapczyk and Przybysz, 2020; Gaido et al., 2021; Shanbhogue et al., 2022). Most recently, Tsiamas et al. (2022b) finetune a wav2vec 2.0 model (Baevski et al., 2020) to predict gold segmentation-like utterance boundaries, an approach which outperforms several alternative segmentation methods and was widely adopted in the 2022 IWSLT offline SLT shared task (Tsiamas et al., 2022a; Pham et al., 2022; Gaido et al., 2022).

Apart from improving automatic audio segmentation methods, previous research has also focused on making SLT models more robust toward segmentation errors. Gaido et al. (2020) and Zhang et al. (2021) both explore context-aware end-to-end SLT models and show that context can help to better translate VAD-segmented utterances. Similarly, training on artificially truncated data can be beneficial to segmentation robustness in cascaded setups (Li et al., 2021) but also in end-to-end models (Gaido et al., 2020). While this approach can introduce misalignments between source audio and target text, such misalignments in the training data are not necessarily harmful to SLT models as Ouyang et al. (2022) recently showed in an evaluation of the MuST-C dataset (Di Gangi et al., 2019).

Both of these approaches – improving automatic segmentation and making models more robust toward segmentation errors – can be combined. For example, Papi et al. (2021) show that continued finetuning on artificial segmentation can help narrow the gap between hybrid segmentation approaches and manual segmentation. However, a combination of both methods is not always equally beneficial. Gaido et al. (2022) repeat Papi et al. (2021)'s analysis with the segmentation model proposed by Tsiamas et al. (2022b) and show that for this segmentation strategy, continued finetuning on resegmented data does not lead to an improvement in translation quality.

In our work, we aim to extend these efforts and test various combinations of segmentation and model finetuning strategies. We are especially interested in fixed-window segmentations which have largely been ignored in SLT research but are attractive from a practical point of view because they do not require an additional model to perform segmentation. To the best of our knowledge, we are the first to perform such an extensive segmentation-

	train		test	
	# talks	# segments	# talks	# segments
en-de	2,043	229,703	27	2,641
es-en	378	36,263	15	996
fr-en	250	30,171	11	1,041
it-en	221	24,576	11	979
pt-en	279	30,855	11	1,022
multi	1,128	121,865	48	4038

Table 1: Overview of dataset statistics. The last row shows the total numbers for the multilingual model on es-en, fr-en, it-en and pt-en combined.

focused analysis for online SLT, considering delay, flicker and translation quality for the evaluation.

3 Experiment Setup

3.1 Data

We run experiments with TED talk data in five different language pairs where the task is to translate a TED talk as an incoming stream without having any gold sentence segmentation.

For English-to-German, we use the data from the MuST-C corpus (Di Gangi et al., 2019) version 1.0[2]. This dataset is built from TED talk audio with human-annotated transcriptions and translations. For testing, we use the "tst-COMMON" test set. For Spanish-, French-, Italian- and Portuguese-to-English, we use the data from the mTEDx corpus (Salesky et al., 2021)[3]. This dataset is also based on TED talks and provides human annotated transcriptions and translations of the audio files. For testing, we use the "iwslt2021" test set from the IWSLT 2021 multilingual speech translation shared task (Anastasopoulos et al., 2021). The dataset statistics can be seen in Table 1.

3.2 Spoken Language Translation Models

We base all our experiments on the joint speech- and text-to-text model (Tang et al., 2021a,b,c) released by Meta AI. For the English-German experiments, we use the model provided by Tang et al. (2021b)[4] and for the other language pairs, we use the multilingual model provided by Tang et al. (2021a)[5]. We refer to these models as the **original** models. These models are trained on full segments that mostly comprise one sentence:

> **And like with all powerful technology, this brings huge benefits, but also some risks.**

To investigate the effects of different segmentation strategies combined with segmentation-robust models, we finetune three different variants based on each model. In each case, the finetuning data is augmented with artificially segmented data, but no segments cross the boundaries between the individual TED talks.

- **prefix:** This model is finetuned on a 50-50 mix of original segments and synthetically created prefixes (i.e. sentences where the end is arbitrarily chopped off). Finetuning on prefixes should help for translating artificially segmented audio where the segment stops in the middle of an utterance. We create prefixes of the original segments by randomly sampling a new duration for an audio segment and using the length ratio to extract the corresponding target text. An example for a prefixed version of the original segment can be seen here:

  > **And like with all**

- **context:** This model is finetuned on a mix of original segments and synthetically created longer segments. Context was already shown to help with segmentation errors by Zhang et al. (2021). This model should be able to translate segments that consist of multiple utterances. For each segment in the original training set, we randomly either use the original segment (50% of the time) or an extended segment created by prepending the previous segment (25% of the time) or the 2 previous segments (also 25% of the time). We then add context-prefixed segments for each of these (possibly-extended) segments, by truncating the last concatenated segment. An example for a context-prefixed version of the original segment can be seen here:

  > We work every day to generate those kinds of technologies, safe and useful. **And like with all powerful technology, this brings huge benefits,**

- **windows:** This model is finetuned on a 50-50 mix of original segments and windows of random duration. We split the audio into windows by starting at the beginning of the audio

[2] https://ict.fbk.eu/must-c/

[3] http://www.openslr.org/100

[4] https://github.com/facebookresearch/fairseq/blob/main/examples/speech_text_joint_to_text/docs/ende-mustc.md

[5] https://github.com/facebookresearch/fairseq/blob/main/examples/speech_text_joint_to_text/docs/iwslt2021.md

and then sampling the duration of the first window. The end of this window then becomes the start of the next window and we repeat this process until we reach the end of a TED talk. For every such window, we extract the corresponding target text from the time-aligned gold segment(s) via length ratios. This mirrors the conditions at inference time with a fixed-window segmentation where a segment can start and end anywhere in an utterance and can also comprise multiple utterances. The segment durations are sampled uniformly between 10 and 30 seconds. Note that this model will see the qualitatively poorest data out of all finetuned models because both the end of the segment and the beginning depend on length ratios which can introduce alignment errors. An example for a window version of the original segment can be seen here:

> or death diagnosis without the help of artificial intelligence. We work every day to generate those kinds of technologies, safe and useful. **And like with all powerful technology, this brings huge benefits, but also some risks.** I don't know how this debate ends, but what I'm sure of, is that the game

All models are trained from the original checkpoint for an additional 20k steps and the last two checkpoints are averaged if more than one is saved. We do this finetuning by continuing training with the config file of the original model. For the English→German MuST-C model, we train on the audio as well as the corresponding phoneme sequences based on the transcript, however, we do not use additional parallel text data during finetuning. For the multilingual mTEDx model, we only train on data for the selected language pairs and only on audio (no phoneme sequences) because this model was already finetuned on the spoken language translation task. The validation sets only contain gold segments and all models stop training due to the step limit before early stopping is triggered.

3.3 Segmentation Strategies

We consider four different inference-time segmentation strategies in our experiments, visualised in Figure 1:

- **gold:** These are human annotated segmentation boundaries that are released as part of the MuST-C and mTEDx data. This segmentation can be viewed as an oracle segmentation even though it may not necessarily be the best segmentation for all models. Using the gold segmentation in practice is unrealistic, especially in the online setting where there would be no time for a human to segment the audio before translation.

- **SHAS:** This segmentation method was recently proposed by Tsiamas et al. (2022b). The authors finetune a pretrained wav2vec 2.0 model (Baevski et al., 2020) on the gold segmentations and train it to predict probabilities for segmentation boundaries. SHAS can be used both in offline and online setups using different algorithms to determine the segmentation boundaries based on the model's probabilities. Since we perform our experiments in an online setup, we use the pSTREAM algorithm to identify segments with SHAS. We set the maximum segment length to 18 seconds which the authors reported as best-performing.

- **fixed:** This is a simple approach that splits the audio stream into independent fixed windows of a given duration. In our experiments, we use durations of 26 seconds, which performed best in experiments by Tsiamas et al. (2022b).

- **merged:** Similarly to above, we consider fixed-size windows for this segmentation strategy but here we construct overlapping windows. We use a duration of 15 seconds[6] and shift the window with a stride of 2 seconds at a time. The translations of these overlapping windows are merged before the next window is translated (see Section 3.5).

3.4 Retranslation

We employ a retranslation strategy (Niehues et al., 2016, 2018; Arivazhagan et al., 2020a,b) for our end-to-end SLT experiments. This means that we retranslate the incoming audio at fixed time intervals. In our experiments, we retranslate every 2 seconds to be consistent with the 2-second stride from the merging windows approach. Because of such retranslations of the full audio segment — from the start of the segment up to the current time step — the SLT model may correct translation mistakes from earlier time steps. This means that the

[6] We found empirically that this works better than a duration of 26 seconds as for fixed-windows, with both increased translation quality and reduced flicker (see Appendix B).

final translation of a complete segment reaches the quality of offline translation. However, if these updated partial translations are presented to users and there are changes to previously translated text, this may be hard to follow. Therefore, it is important to not only evaluate the quality of the translations and the delay but also how often previously translated words are changed which is termed "flicker". Typically, when delay improves there will be more flicker because translating sooner means a higher chance of errors that need to be corrected in the next retranslation.

3.5 Window Merging Algorithm

One reason why a fixed-window segmentation might underperform compared to other segmentations is that utterances are likely to be split up into two or multiple segments which can introduce ambiguities and result in disfluent translations. However, this problem can be reduced if the windows are overlapping which is technically very easy to do. With a retranslation approach, we can simply shift the whole window by X seconds to obtain overlapping translations.

To merge the resulting translations, we employ a merging algorithm that was previously proposed for a cascaded SLT setup (Sen et al., 2022). Their merging window algorithm also works for end-to-end SLT because it is not dependent on a transcript of the source audio. The algorithm identifies the longest common substring (LCS) between the growing translation of the output stream and the translation of the current window. The current output is formed by everything to the left of the LCS coming from the output at the previous time step, followed by the LCS and then everything to the right of the LCS from the current translation output. In this way, the translation of the input stream is continuously extended.

The merging is controlled by a threshold that defines the minimum required length of the LCS. At every time step, this threshold is computed by:

$$threshold = |T_t| * \tau$$

Where T_t is the current window translation length and τ is a ratio hyperparameter. If the LCS is shorter than this minimum length, instead of merging the current translation with the output stream, the window is backtracked to the left and a longer window is translated. We backtrack 0.1 seconds at a time for a maximum of three backtracks. Only when a sufficiently long LCS is found or the maximum number of backtracks is reached, do we perform the merging operation. In our experiments, we set the ratio τ to 0.4 which performed best in the cascaded setup (Sen et al., 2022). If there are multiple LCS (common substrings with the same length), we merge at the last-occurring one.

3.6 Evaluation

For evaluation, we use SLTev[7] (Ansari et al., 2021), a toolkit that can evaluate translation quality, delay and flicker in a retranslation SLT setup. We explain below how the evaluation is adapted for unsegmented input. Since we assume our input is segmented at the talk level, we evaluate at the talk level too.

For **translation quality**, SLTev internally resegments the translations and aligns the new segments to the reference segments such that the word error rate is minimised (Matusov et al., 2005). It is not guaranteed that the new segments follow the sentence boundaries and are perfectly aligned but, as long as the introduced alignment errors are similar for different segmentations, they can be compared.

For **flicker**, we cannot use the sentence-level measure in SLTev because this is computed as an average over all segment-level flicker scores, and with different segmentations, this measure is not comparable. However, the document-level measure is evaluated independent of the segmentation and this works well for our purpose.

For **delay**, we do not use the official implementation in SLTev because of the way it assigns timestamps to repeated tokens. To explain the problem, consider the following example:

P	13.18	O
P	14.18	O horror,
P	15.18	O horror, terror, horror
C	16.18	O horror, horror, horror.

where we retranslate the newly available audio every second and consequently get three partial translations (P) and one final, complete translation (C). In SLTev, every token is assigned the time stamp of its type's first occurrence. This results in the following time stamp assignments with the original implementation.

<table>
<tr><td>O</td><td>horror</td><td>,</td><td>horror</td><td>,</td><td>horror</td><td>.</td></tr>
<tr><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>13.18</td><td>14.18</td><td>14.18</td><td>14.18</td><td>14.18</td><td>14.18</td><td>16.18</td></tr>
</table>

[7] https://github.com/ELITR/SLTev

All occurrences of “horror” and “,” are assigned the timestamp 14.18 even though most of them are not yet generated by that time. If we translate longer segments that may be comprised of multiple sentences, encountering tokens that were already seen before becomes more and more likely. All of those would be assigned the timestamp of the first occurrence which favours longer segments (which we take to the extreme with our merged windows output stream). To solve this issue, we adapt the delay computation and store the individual timestamps for all repeated tokens. For this, we also need to be aware that previous content can change with each retranslation (e.g. terror → horror). We solve this following Arivazhagan et al. (2020b)’s notation of content delay and only assign timestamps once the previous context has finalised:

O	horror	,	horror	,	horror	.
|	|	|	|	|	|	|
13.18	14.18	14.18	16.18	16.18	16.18	16.18

With these new timestamps, all possible segmentations will receive the same delay if the translated text is identical and longer segments are no longer favoured in the SLTev delay calculation. However, since we wait until the context has finalised before we assign the time stamps, the new delay measure is now also affected by flicker.

4 Results

4.1 Translation Quality

We compare the different SLT models on different segmentations of the test sets and show the resulting translation quality of the complete segments in terms of BLEU in Table 2. Note that we would reach the same translation quality in an offline setting because the final retranslation is a translation of the full window, and in common with previous work, translation quality of online SLT is only measured on the final retranslation. We also evaluate with COMET (Rei et al., 2020) and report even better results with the merging windows approach but also find that COMET might be less reliable in a streaming SLT setup due to resegmentation errors (see Section D.1).

Does SHAS perform best with the original model (first column) as in previous work? When the SLT model is just trained on gold data, SHAS proves to be the best-performing segmentation out of all automatic segmentations which is in line with results by Tsiamas et al. (2022b) and Gaido et al. (2022). As in previous studies, we also find that the original model shows a considerable drop in BLEU when moving from the gold segmentation to automatically split segments.

		original	prefix	context	window
	gold	25.4	25.5	25.2	25.5
en-de	SHAS	24.5	23.9	24.9	24.8
	fixed	22.4	21.1	23.6	23.1
	merged	24.8	23.8	**25.3**	22.8
	gold	41.6	41.3	41.1	41.4
es-en	SHAS	40.2	40.3	40.7	41.0
	fixed	35.0	36.9	39.6	38.4
	merged	38.9	39.9	**42.0**	39.7
	gold	37.2	36.2	35.6	35.6
fr-en	SHAS	**36.2**	36.1	35.8	36.1
	fixed	31.0	32.0	34.5	32.9
	merged	34.6	35.2	35.8	31.9
	gold	27.0	28.7	28.8	29.0
it-en	SHAS	26.4	28.0	28.7	29.0
	fixed	22.5	25.6	27.5	26.3
	merged	25.3	27.4	**29.2**	27.6
	gold	30.6	29.5	28.7	29.1
pt-en	SHAS	**29.5**	28.9	29.2	28.6
	fixed	23.6	24.0	26.9	26.2
	merged	26.6	27.4	28.1	24.3

Table 2: BLEU scores with different SLT models (columns) and different audio segmentation methods (rows). Best result for *automatic* segmentation scenario marked in bold and green.

Is SHAS still the best-performing segmentation with the finetuned models? Finetuning with alternative segmentations can offer strong improvements for SHAS (+2.3) on it-en, with small improvements on es-en and en-de, but lower BLEU on pt-en and fr-en. Similarly, Gaido et al. (2022) found that SHAS did not benefit from finetuning on resegmented data. However, for the two segmentation approaches based on fixed windows, finetuning greatly reduces the gap to the gold segmentation.

This is especially noticeable when we finetune on context and prefixes (third column). This confirms the finding by Zhang et al. (2021) that context-aware models can better translate artificially segmented audio. When merging overlapping windows, we consistently see an improvement over the segmentation with non-overlapping fixed windows. In three language pairs, this method outperforms SHAS and in two the context-finetuned model even improves over the gold segmentation.

Do training conditions need to match the segmentation at inference time? Apart from the context-aware finetuned model, we also finetuned a model on fixed windows of random duration (last column). This matches the fixed-window audio input at inference time better because a segment can start anywhere in an utterance, unlike the context-based model where every training segment started at the beginning of an utterance. Surprisingly, we find that the model finetuned on windows of random duration generally performs worse with the merging window strategy than the context-based model. This suggests that the training data for this model contains more misalignments between speech and translation because we extract both the start and the end of the segment via length ratios. This causes more flicker (see next Section) which makes it harder to merge the translations at each time step correctly. We leave extended experiments where the alignments between speech and translation are computed via ASR or the SLT output of the windows of random durations (as opposed to a simple length ratio) to future work.

4.2 Flicker

As mentioned in the Introduction, translation quality is not the only important evaluation metric in an online SLT scenario. When using a retranslation approach, we also need to consider the flicker that is caused by the model updating its translations at every time step. We compute the flicker as described in Section 3.6. The flicker scores for the Spanish-to-English test set can be seen in Figure 2, the same figures for the other language pairs are in Appendix D. For the results shown here, we use an output mask of 0. We show in Appendix C that our findings also hold with larger output masks. We show scores with and without biased beam search.

Biased beam search (Arivazhagan et al., 2020a) is a modification to regular beam search that biases the probability distribution at the current time step towards a token in a given prefix translation at the same timestep. This can be used to stabilise retranslation – the translation of the current prefix is biased towards the translation of the previous prefix, suppressing flicker. In our experiments, we use the translation of the previous step as the prefix with a beta parameter of 0.25 and mask the 5 last tokens such that changes towards the end of the sentence are still possible[8]. Biased beam search cannot be applied directly to the merged window approach, since it depends on an alignment between the translation of the current prefix and that of the previous prefix. When translating using sliding windows, the current and previous prefixes have different start points, so their translations cannot be easily aligned. We experimented with a way to reduce flicker by merging on the last common substring rather than the longest but this causes considerable translation quality loss (see Appendix B).

Does the segmentation strategy matter for flicker? From Figure 2, we can see that there are big differences between the different segmentation strategies. Fixed windows have the highest flicker because there we translate the longest windows. If something at the beginning of the window translation is changed, this will increase the flicker score considerably. With biased beam search, the flicker can be dramatically reduced. Merging overlapping windows has a lower flicker than fixed windows without biased beam search, both because the duration of the windows is shorter and because the merging algorithm prohibits changes to the left of the longest common substring[9]. This segmentation method even has lower flicker than SHAS when no biased beam search is applied. With biased beam search, SHAS performs mostly similar to the gold segmentation which has the lowest flicker overall.

Does model finetuning help reduce flicker? Prefix finetuning helps reduce flicker both with and without biased beam search because the models see incomplete sentences at training time and are less likely to hallucinate to finish the sentence. Context finetuning helps even more and we saw in the outputs that this model has less of a tendency to connect multiple sentences into a longer sentence which can reduce flicker. The model finetuned on windows shows an even higher flicker than the original model for most segmentation strategies even though it was designed to be able to translate seg-

[8] We do not show translation quality scores with biased beam search because on average there is only a difference of -0.006 BLEU.

[9] Reducing the window length to 15 seconds for the fixed window segmentation reaches a flicker that is only slightly higher than for merged windows but the translation quality suffers considerably.

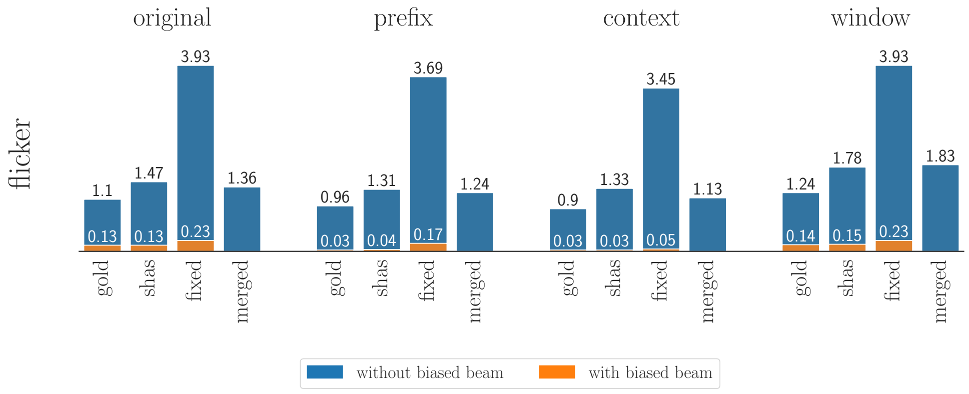


Figure 2: Flicker values for the different segmentation strategies and SLT models on the Spanish-to-English test set. The results are grouped by training strategy and each bar corresponds to a different segmentation strategy. We do not apply biased beam search to the merged segmentation.

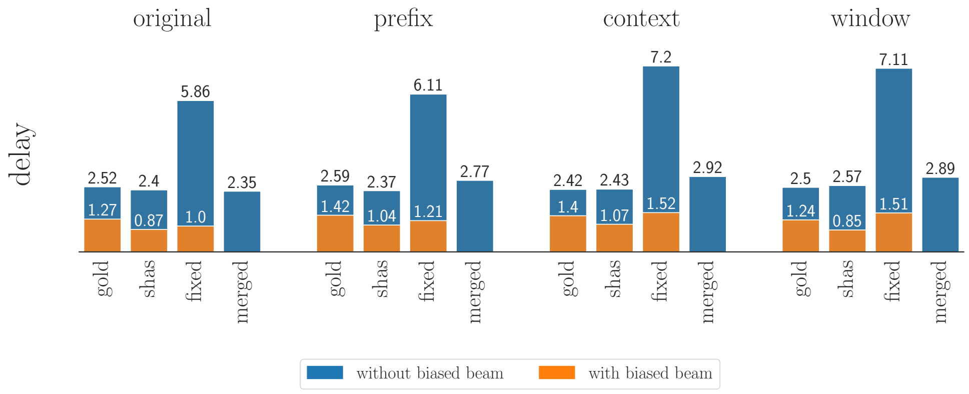


Figure 3: Delay values for the different segmentation strategies and SLT models on the Spanish-to-English test set. The results are grouped by training strategy and each bar corresponds to a different segmentation strategy.

ments that can start and end anywhere in a sentence. As discussed in the previous section, we think this increased flicker is an artefact of the automatically generated training data which can be erroneous.

4.3 Delay

The final evaluation metric we consider is delay. The results can be seen in Figure 3. Again, we show results with and without biased beam search for the gold, SHAS and fixed-window segmentation.

Does the segmentation strategy matter for delay? Because our definition of delay is affected by flicker as well (see Section 3.6), the fixed segmentation without biased beam search not only has the highest flicker but also the highest delay. In our results, we can see that the high delay is caused by the flicker because when we reduce flicker with biased beam search the fixed segmentation has comparable delay to the gold and SHAS segmentations. The merging windows approach has comparable delay to the gold and SHAS segmentations without biased beam search. Since we cannot apply biased beam search reliably to the merging windows approach without hurting translation quality, the flicker cannot be reduced and therefore, the merging windows approach has higher delay than the other segmentation methods with biased beam search. If delay could be defined independently of flicker in a way that still works for comparing different segmentations, the merging windows approach would likely have similar delay also compared to the outputs with biased beam search.

Does model finetuning help reduce delay? The results are a bit mixed. For example, the context model reduces delay for the gold segmentation but increases it slightly for SHAS and more for the fixed segmentation and the merging windows approach. In general, the choice of the model does not seem to be as important for delay as for translation quality and to a lesser extent flicker. It is possible that apparent effects only occur because our definition of delay is affected by flicker.

5 Discussion

Based on our results in Section 4, we believe that fixed-window segmentation should not be disregarded in future SLT research on unsegmented audio. Given the right setup with a context-aware model and a merging window algorithm, this segmentation can outperform current state-of-the-art automatic segmentation models and in some cases even the gold segmentation in terms of translation quality. Moreover, in an online SLT setup, a fixed-window approach brings the additional benefit that no dedicated segmentation model needs to be loaded at inference time and run every time new audio becomes available.

While there is currently no solution to bring flicker down to biased beam search levels without hurting quality (see Appendix B) or increasing delay (see Appendix C), this should not be a reason to disregard fixed-window segmentation as it opens exciting opportunities for future research.

6 Conclusion

In this paper, we explored several combinations of segmentation-robust finetuning and different automatic segmentation strategies in an online SLT setup. We focus on a retranslation-based approach to SLT and we run experiments on five different language pairs based on two different SLT models. Considering the evaluation of translation quality, flicker and delay, we discuss several issues that arise when comparing different segmentations and propose a fix to an existing toolkit for evaluating delay. Our results show that a simple fixed-window segmentation can perform surprisingly well if an algorithm is used for merging overlapping windows and a context-aware SLT model is used. In terms of translation quality, this segmentation performs comparably to SHAS — the current state-of-the-art segmentation method — and in some cases even outperforms the gold segmentation, showing potential for future application to offline SLT. In terms of flicker and delay, the results of the merging windows approach are comparable to the other segmentations if biased beam search is not enabled but future work is needed to reduce flicker in the merging windows approach to similar levels as biased beam search for other strategies without hurting translation quality.

Ethical Considerations

In our work, we only use publicly available model checkpoints, toolkits and datasets and do not collect any additional data. Our experiments also do not involve human annotators.

Limitations

While we aim to evaluate on a number of language pairs and with different automatic metrics, there are still some open questions that we could not answer in this work. First, we did not perform a human evaluation and, therefore, it remains unclear how distracting the different flicker and delay values with different setups would be for a user. However, previous work by Macháček and Bojar (2020) shows that character erasure - a metric related to flicker - correlates with usability scores in a human evaluation which suggests that this would also be true for flicker. Second, the current implementation of SHAS can be used to simulate an online setting but it still expects the full audio as input. Consequently, we could not empirically compare how long translation takes with different segmentation methods in a real online setup. Third, our experiments are limited to SLT using a retranslation strategy. We leave further experiments with simultaneous SLT models that use a policy to decide at each time step whether to wait for further input or to translate for the future.

Acknowledgements

We thank Sukanta Sen and Ioannis Tsiamas who shared their code with us for the window merging algorithm and the pSTREAM implementation of SHAS, respectively. We are also grateful to Noëmi Aepli and Amit Moryossef and the anonymous reviewers for their valuable feedback. This work has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreements (project ELITR; no. 825460 and project GoURMET; no. 825299), and from the Swiss National Science Foundation (project MUTAMUR; no. 176727). All experiments made use of the infrastructure services provided by S3IT, the Service and Support for Science IT team at the University of Zurich.

References

Antonios Anastasopoulos, Loïc Barrault, Luisa Bentivogli, Marcely Zanon Boito, Ondřej Bojar, Roldano

Cattoni, Anna Currey, Georgiana Dinu, Kevin Duh, Maha Elbayad, Clara Emmanuel, Yannick Estève, Marcello Federico, Christian Federmann, Souhir Gahbiche, Hongyu Gong, Roman Grundkiewicz, Barry Haddow, Benjamin Hsu, Dávid Javorský, Věra Kloudová, Surafel Lakew, Xutai Ma, Prashant Mathur, Paul McNamee, Kenton Murray, Maria Nădejde, Satoshi Nakamura, Matteo Negri, Jan Niehues, Xing Niu, John Ortega, Juan Pino, Elizabeth Salesky, Jiatong Shi, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Yogesh Virkar, Alexander Waibel, Changhan Wang, and Shinji Watanabe. 2022. Findings of the IWSLT 2022 evaluation campaign. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 98–157, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Antonios Anastasopoulos, Ondřej Bojar, Jacob Bremerman, Roldano Cattoni, Maha Elbayad, Marcello Federico, Xutai Ma, Satoshi Nakamura, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Alexander Waibel, Changhan Wang, and Matthew Wiesner. 2021. FINDINGS OF THE IWSLT 2021 EVALUATION CAMPAIGN. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 1–29, Bangkok, Thailand (online). Association for Computational Linguistics.

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondřej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, Alexander Waibel, and Changhan Wang. 2020. FINDINGS OF THE IWSLT 2020 EVALUATION CAMPAIGN. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 1–34, Online. Association for Computational Linguistics.

Ebrahim Ansari, Ondřej Bojar, Barry Haddow, and Mohammad Mahmoudi. 2021. SLTEV: Comprehensive evaluation of spoken language translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 71–79, Online. Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, Te I, Wolfgang Macherey, Pallavi N. Baljekar, and George F. Foster. 2020a. Re-translation strategies for long form, simultaneous, spoken language translation. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7919–7923.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, and George Foster. 2020b. Re-translation versus streaming for simultaneous translation. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 220–227, Online. Association for Computational Linguistics.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Eunah Cho, Jan Niehues, and Alex Waibel. 2017. NMT-Based Segmentation and Punctuation Insertion for Real-Time Spoken Language Translation. *Proceedings of Interspeech*, pages 2645–2649.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.

Marco Gaido, Mattia A. Di Gangi, Matteo Negri, Mauro Cettolo, and Marco Turchi. 2020. Contextualized Translation of Automatically Segmented Speech. In *Proc. of Interspeech 2020*, pages 1471–1475.

Marco Gaido, Matteo Negri, Mauro Cettolo, and Marco Turchi. 2021. Beyond voice activity detection: Hybrid audio segmentation for direct speech translation. In *Proceedings of the Fourth International Conference on Natural Language and Speech Processing (ICNLSP 2021)*, pages 55–62, Trento, Italy. Association for Computational Linguistics.

Marco Gaido, Sara Papi, Dennis Fucci, Giuseppe Fiameni, Matteo Negri, and Marco Turchi. 2022. Efficient yet competitive speech translation: FBK@IWSLT2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 177–189, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Daniel Li, Te I, Naveen Arivazhagan, Colin Cherry, and Dirk Padfield. 2021. Sentence boundary augmentation for neural machine translation robustness. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Dominik Macháček and Ondřej Bojar. 2020. Presenting simultaneous translation in limited space. In *Proceedings of the 20th Conference Information Technologies - Applications and Theory (ITAT 2020)*.

Evgeny Matusov, Gregor Leusch, Oliver Bender, and Hermann Ney. 2005. Evaluating machine translation output with automatic sentence segmentation. In *Proceedings of the Second International Workshop on Spoken Language Translation*, Pittsburgh, Pennsylvania, USA.

Jan Niehues, Thai Son Nguyen, Eunah Cho, Thanh-Le Ha, Kevin Kilgour, Markus Müller, Matthias Sperber, Sebastian Stüker, and Alexander H. Waibel. 2016. Dynamic transcription for low-latency speech translation. In *Proceedings of Interspeech*.

Jan Niehues, Ngoc-Quan Pham, Thanh-Le Ha, Matthias Sperber, and Alex Waibel. 2018. Low-Latency Neural Speech Translation. In *Annual Conference of the International Speech Communication Association (InterSpeech)*, Hyderabad, India.

Siqi Ouyang, Rong Ye, and Lei Li. 2022. On the impact of noises in crowd-sourced data for speech translation. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 92–97, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2021. Dealing with training and test segmentation mismatch: FBK@IWSLT2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 84–91, Bangkok, Thailand (online). Association for Computational Linguistics.

Ngoc-Quan Pham, Tuan Nam Nguyen, Thai-Binh Nguyen, Danni Liu, Carlos Mullov, Jan Niehues, and Alexander Waibel. 2022. Effective combination of pretrained models - KIT@IWSLT2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 190–197, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Tomasz Potapczyk and Pawel Przybysz. 2020. SR-POL's system for the IWSLT 2020 end-to-end speech translation task. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 89–94, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Elizabeth Salesky, Matthew Wiesner, Jacob Bremerman, Roldano Cattoni, Matteo Negri, Marco Turchi, Douglas W. Oard, and Matt Post. 2021. Multilingual tedx corpus for speech recognition and translation. In *Proceedings of Interspeech*.

Sukanta Sen, Ondřej Bojar, and Barry Haddow. 2022. Simultaneous translation for unsegmented input: A sliding window approach. arXiv preprint 2210.09754.

Akshaya Shanbhogue, Ran Xue, Ching-Yun Chang, and Sarah Campbell. 2022. Amazon Alexa AI's system for IWSLT 2022 offline speech translation shared task. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 169–176, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Jongseo Sohn, Nam Soo Kim, and Wonyong Sung. 1999. A statistical model-based voice activity detection. *IEEE Signal Processing Letters*, 6(1):1–3.

Yun Tang, Hongyu Gong, Xian Li, Changhan Wang, Juan Pino, Holger Schwenk, and Naman Goyal. 2021a. FST: the FAIR speech translation system for the IWSLT21 multilingual shared task. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 131–137, Bangkok, Thailand (online). Association for Computational Linguistics.

Yun Tang, Juan Pino, Xian Li, Changhan Wang, and Dmitriy Genzel. 2021b. Improving speech translation by understanding and learning from the auxiliary text translation task. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4252–4261, Online. Association for Computational Linguistics.

Yun Tang, Juan Miguel Pino, Changhan Wang, Xutai Ma, and Dmitriy Genzel. 2021c. A general multi-task learning framework to leverage text data for speech to text tasks. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6209–6213.

Ioannis Tsiamas, Gerard I. Gállego, Carlos Escolano, José Fonollosa, and Marta R. Costa-jussà. 2022a. Pretrained speech encoders and efficient fine-tuning methods for speech translation: UPC at IWSLT 2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 265–276, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, and Marta R. Costa-jussà. 2022b. Shas: Approaching optimal segmentation for end-to-end speech translation. In *Proceedings of Interspeech*.

Biao Zhang, Ivan Titov, Barry Haddow, and Rico Sennrich. 2021. Beyond sentence-level end-to-end speech translation: Context helps. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2566–2578, Online. Association for Computational Linguistics.

Appendix

A Further Finetuning Specifications

We finetune all models and translate with a single NVIDIA Tesla V100 GPU. For the multilingual mTEDx model, the additional parameter `load-speech-only` needs to be added to the official training script[10]. We use the `restore-file` parameter to specify the checkpoints of the original models from which continued training should be initialised.

We will release all code (training scripts, translation scripts and evaluation modifications), the finetuned model checkpoints and the outputs upon publication.

B Experiments with Last Common Subsequence

As a possible way of reducing flicker for the merging windows approach, we try merging on the last common subsequence (longer than two tokens) instead of the longest common subsequence. In this way, we can maximise the finalised part of the growing output translation and reduce flicker. Figure 4 shows how the flicker increases for both merging strategies when the window size increases. With the original implementation that merges on the longest common subsequence, the flicker increases dramatically when the window size is increased. For the modified merging algorithm that merges on the last subsequence (longer than two tokens) the flicker increases only moderately with increased window size and is in general much lower.

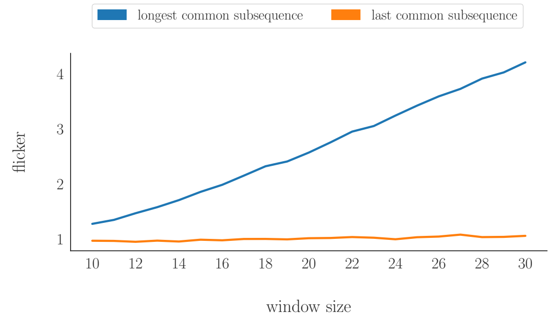


Figure 4: Flicker values with the original model on the English-to-German test set for different window sizes when merging on the longest common subsequence (blue) and the last common subsequence (orange).

Based on these results, one might choose to merge on the last sequence, however, this change also affects the translation quality. Figure 5 shows the BLEU scores of both merging methods with different window sizes. Unfortunately, merging on the last common subsequence performs continuously worse than merging on the longest common subsequence. If quality is the main focus, this merging method is not advisable. These results also show that a window size of 15 performs best for the merging windows approach.

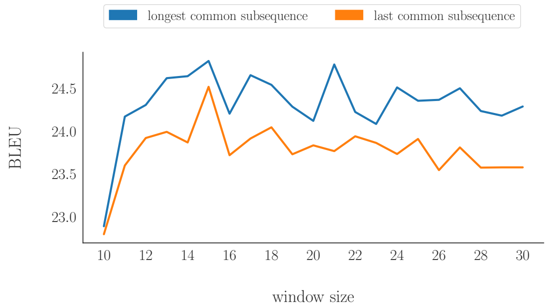


Figure 5: BLEU scores with the original model on the English-to-German test set for different window sizes when merging on the longest common subsequence (blue) and the last common subsequence (orange).

C Results with Output Mask

We also evaluate the four different segmentation methods when an output mask is applied. This means at every time step the output is truncated from the right. The number of tokens that are removed is defined by the mask size, i.e. a mask of size 0 means no tokens are removed and a mask of size 7 means seven tokens are removed. We compute these results for Spanish-to-English without biased beam search and the context-aware model which showed the lowest flicker in general.

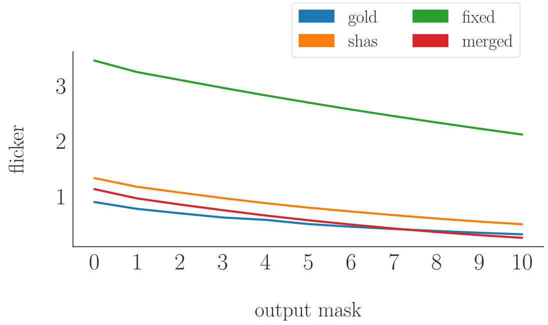


Figure 6: Flicker with different output masks on the Spanish-to-English test set. Results for all four segmentation methods with the context-finetuned model.

[10] https://github.com/facebookresearch/fairseq/blob/main/examples/speech_text_joint_to_text/docs/iwslt2021.md

Figure 6 shows the flicker at different output mask sizes. First of all, it can be noticed that the fixed window segmentation has a continuously higher flicker than all other segmentation methods and that the flicker is still rather large even with a mask of size 10. This suggests that most flicker in the fixed-window segmentation does not occur towards the end of the segments.

The merging windows approach consistently has lower flicker than SHAS and with larger mask sizes even lower flicker than the gold segmentation. With a mask of size 10, the flicker is at 0.25 which is comparable to the flicker of the original model with fixed window segmentation where biased beam search is enabled.

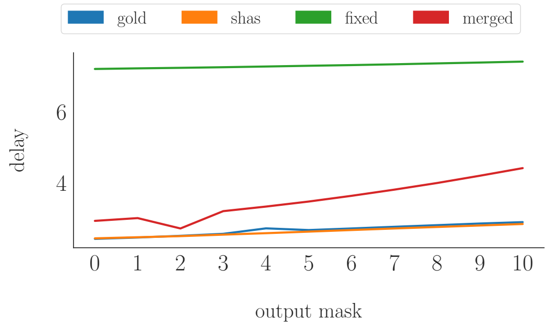


Figure 7: Delay with different output masks on the Spanish-to-English test set. Results for all four segmentation methods with the context-finetuned model.

However, this reduced flicker comes at the cost of a higher delay because the masked tokens will not be available at the time they are actually produced. This flicker-delay trade-off is well-known. Figure 7 shows the increase in delay with larger output masks. For the merging windows approach, we see that the delay increases more than for SHAS and the gold segmentation. Since our definition of the delay measure is affected by flicker, these results are hard to interpret. Nevertheless, using an output mask is a way to reduce flicker for the merging windows approach without reducing translation quality but we need to accept a higher delay.

D Additional Results

D.1 Translation Quality with COMET

For completeness, we present performance results measured with COMET (Rei et al., 2020) in Table 3. This is evaluated outside of SLTev but we use the same resegmentation tool (Matusov et al., 2005) to align the translations with the reference segments. The results show similar patterns as with BLEU and the context model paired with the merging window approach performs best among the automatic segmentation approaches on all language pairs. This approach even outperforms the gold segmentation on three language pairs. Note however that evaluating resegmented text with COMET may have some undesirable side-effects because the translated text is not always split at correct segmentation boundaries, e.g. the first token of a segment often is glued to the end of the previous segment.

		original	prefix	context	window
en-de	gold	-0.0589	-0.0801	-0.0659	-0.0696
	SHAS	-0.1762	-0.1934	-0.0835	-0.1418
	fixed	-0.3080	-0.3655	-0.1846	-0.1671
	merged	-0.1821	-0.2169	**-0.0683**	-0.1133
es-en	gold	0.3175	0.2864	0.2776	0.2981
	SHAS	0.2145	0.2291	0.2784	0.2638
	fixed	-0.0339	0.0448	0.2637	0.2633
	merged	0.2658	0.2736	**0.3962**	0.3642
fr-en	gold	0.1702	0.1380	0.1123	0.1078
	SHAS	0.1147	0.1421	0.1742	0.134
	fixed	-0.1696	-0.1115	0.0777	0.0316
	merged	0.0978	0.1066	**0.2170**	0.1109
it-en	gold	0.0566	0.0583	0.0704	0.0886
	SHAS	-0.012	0.0215	0.0915	0.0709
	fixed	-0.305	-0.2072	0.0142	-0.0408
	merged	-0.0536	-0.0066	**0.1255**	0.0619
pt-en	gold	0.0662	0.0234	-0.0130	-0.0048
	SHAS	-0.0108	-0.0104	-0.0085	-0.0085
	fixed	-0.2939	-0.2853	-0.0854	-0.0937
	merged	-0.0581	-0.0784	**0.0276**	-0.0554

Table 3: COMET scores with different SLT models (columns) and different audio segmentation methods (rows). Best result for *automatic* segmentation scenario marked in bold and green.

We tested this with 200 gold segments for en-de and manually corrected the resegmentation of the original model output. While the BLEU score does not change much with these corrections (24.70 vs. 24.73), the COMET score jumps from -0.1128 to 0.0467 which is a larger improvement than some differences in Table 3. Since it is unclear if such resegmentation errors occur equally often in all our experiment setups, we only include the results with BLEU in the main body of the paper. We hypothesise that COMET has only seen well-formed sentences at training time and consequently is less reliable on such resegmented data. In the future, document-level neural evaluation metrics could be better suited for evaluating translations of unsegmented or automatically segmented audio in SLT.

D.2 Flicker Results for Other Language Pairs

We present the same plots as in Section 4.2 for English-to-German in Figure 8, French-to-English in Figure 9, Italian-to-English in Figure 10 and Portuguese-to-English in Figure 11. The results follow the same patterns as the results for Spanish-English discussed in Section 4.2:

- Fixed windows without biased beam search have the highest flicker.
- For the language pairs into English, the merging windows approach has lower flicker than SHAS if no biased beam search is used.
- Finetuning on context reduces flicker.

D.3 Delay Results for Other Language Pairs

We present the same plots as in Section 4.3 for English-to-German in Figure 12, French-to-English in Figure 13, Italian-to-English in Figure 14 and Portuguese-to-English in Figure 15. The results follow the same patterns as the results for Spanish-English discussed in Section 4.3:

- Fixed windows without biased beam search have the highest delay.
- The merging windows approach has comparable delay to SHAS if no biased beam search is used.
- Finetuning has less of an effect on delay than on flicker.

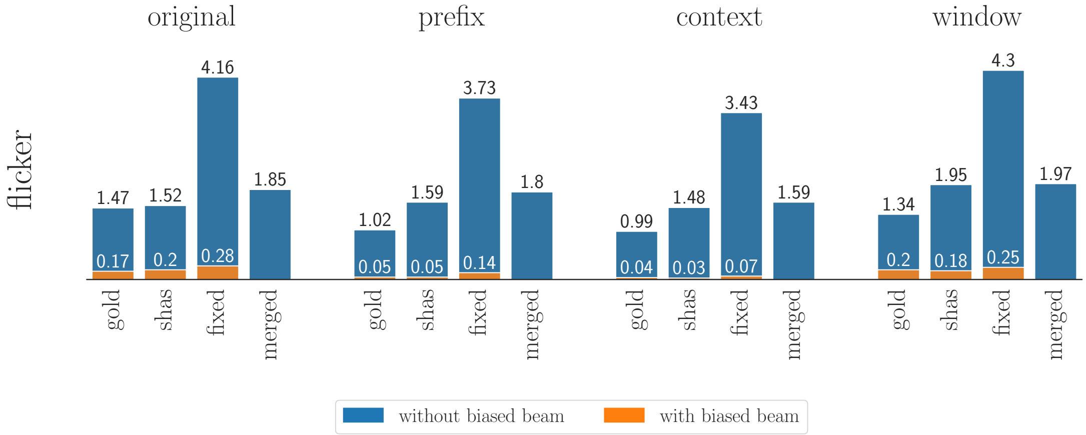


Figure 8: Flicker values for the different segmentation strategies and SLT models on the English-to-German test set.

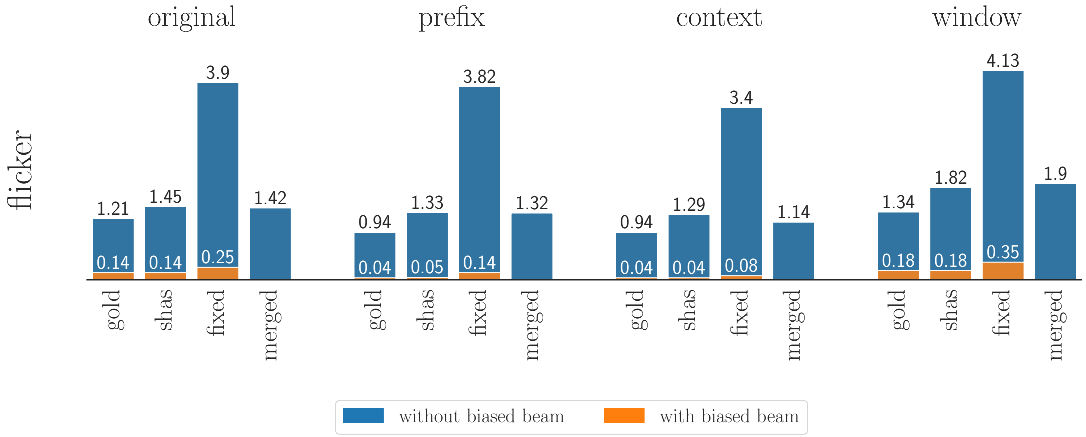


Figure 9: Flicker values for the different segmentation strategies and SLT models on the French-to-English test set.

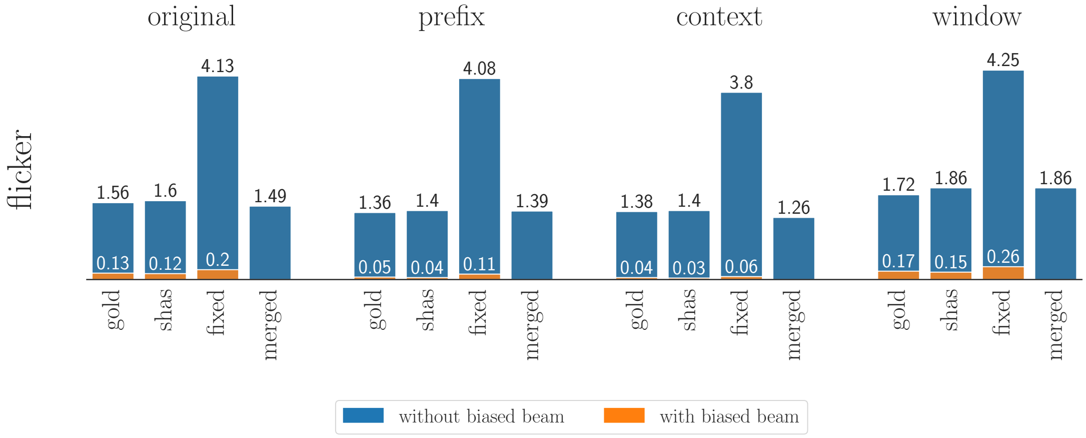


Figure 10: Flicker values for the different segmentation strategies and SLT models on the Italian-to-English test set.

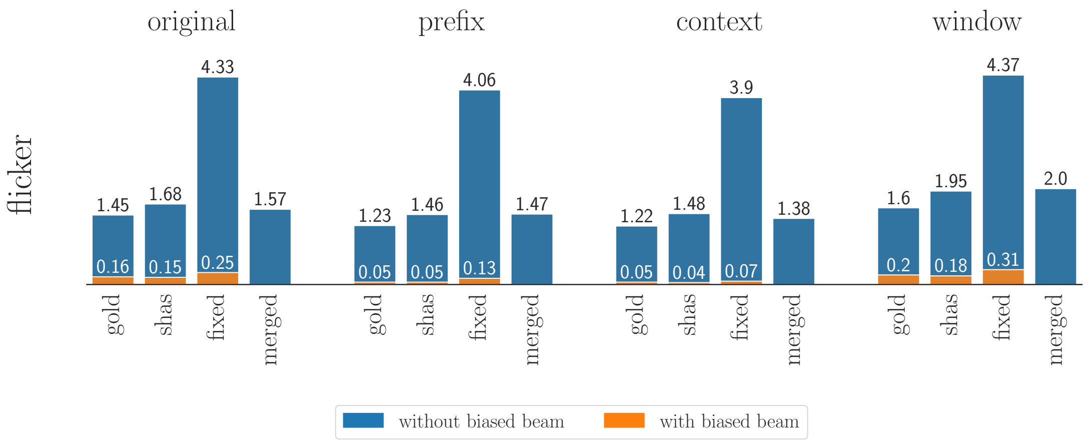


Figure 11: Flicker values for the different segmentation strategies and SLT models on the Portuguese-to-English test set.

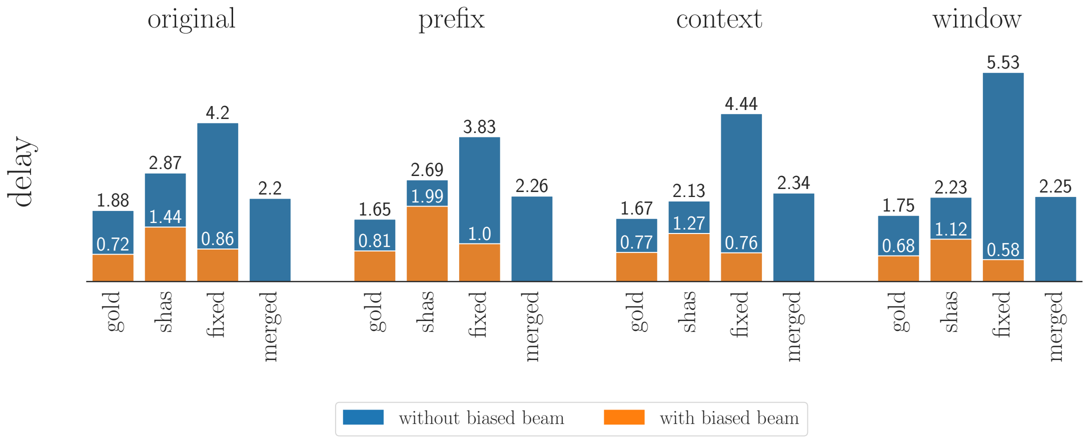


Figure 12: Delay values for the different segmentation strategies and SLT models on the English-to-German test set.

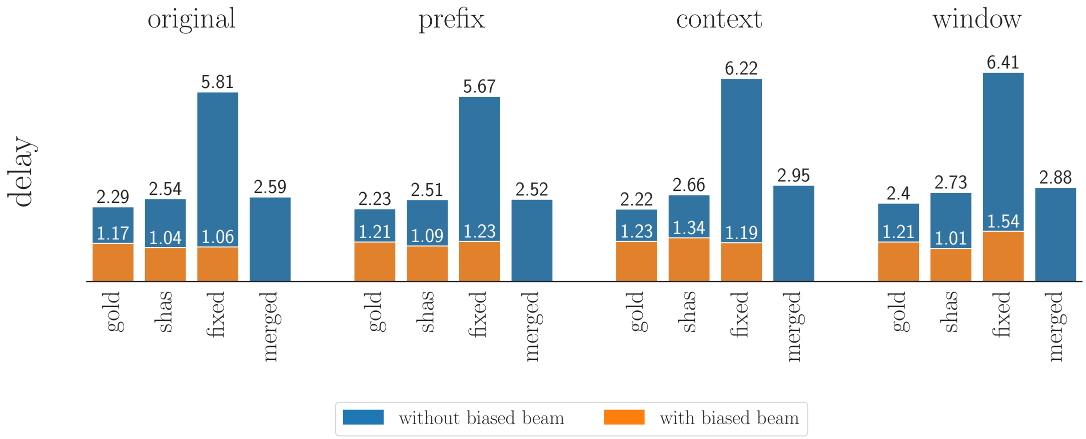


Figure 13: Delay values for the different segmentation strategies and SLT models on the French-to-English test set.

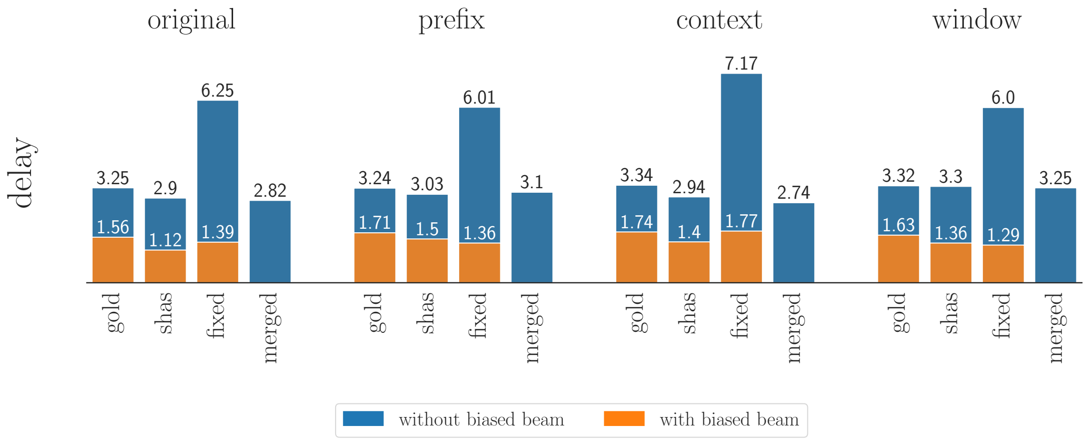


Figure 14: Delay values for the different segmentation strategies and SLT models on the Italian-to-English test set.

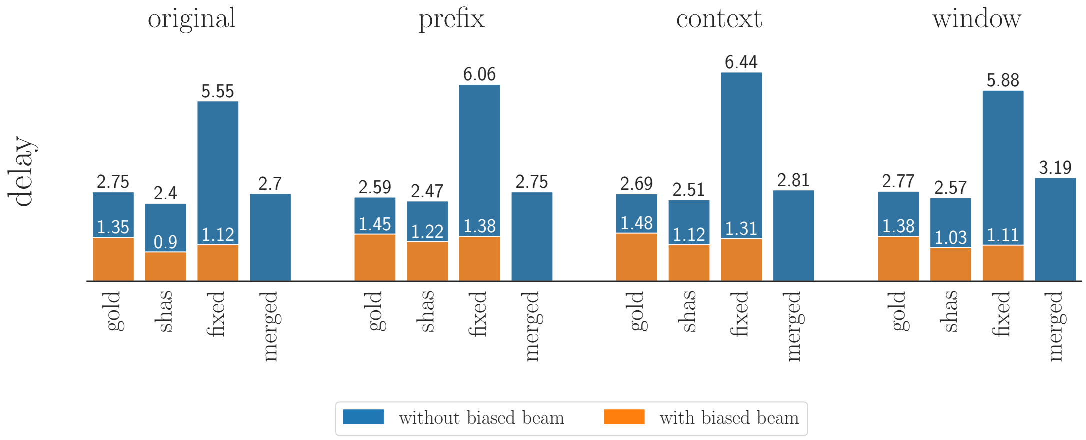


Figure 15: Delay values for the different segmentation strategies and SLT models on the Portuguese-to-English test set.

Additive Interventions Yield Robust Multi-Domain Machine Translation Models

Elijah Rippeth[*]
Department of Computer Science
University of Maryland
erip@cs.umd.edu

Matt Post
Microsoft
mattpost@microsoft.com

Abstract

Additive interventions are a recently-proposed mechanism for controlling target-side attributes in neural machine translation. In contrast to tag-based approaches which manipulate the raw source sequence, interventions work by directly modulating the encoder representation of all tokens in the sequence. We examine the role of additive interventions in a large-scale multi-domain machine translation setting and compare its performance in various inference scenarios. We find that while the performance difference is small between intervention-based systems and tag-based systems when the domain label matches the test domain, intervention-based systems are robust to label error, making them an attractive choice under label uncertainty. Further, we find that the superiority of single-domain fine-tuning comes under question when training data size is scaled, contradicting previous findings.

1 Introduction

Multi-domain machine translation (MDMT) is the paradigm in which a single model is trained to service many domains by training on multiple corpora covering disparate labeled domains. The goal of MDMT is not only to provide high quality *general* machine translation enabled by knowledge transfer across domains, but also to enable high quality *domain-specific* machine translation when a model is provided cues about the target domain, used to control the generation. Though an intuitive task, the expectations surrounding the task were only recently formalized by Pham et al. (2021) in which the authors provided both a set of functional requirements demanded of successful MDMT models and an experimental framework under which those requirements can be tested.

Pham et al. (2021) explored several mechanisms for controlling domain, ranging from simple tag-based approaches to meta-learning based mechanisms. According to the functional requirements outlined by the authors, no method meets all the expectations demanded of effective multi-domain machine translators, though the experiments were run on a relatively small dataset of only in-domain data. The primary remaining expectations, according to the authors, are the superiority of fine-tuning based methods as compared to these methods which can control the target domain, and the ability to accommodate fuzzy or uncertain domains.

This framework is useful, but the authors leave open several other questions regarding the state of MDMT. The first of these is data size. Previous experiments focused only on relatively small, in-domain data in an otherwise high-resource setting of English-French and found that most models pale in comparison to models fine-tuned on a single domain. We wonder whether this fine-tuning superiority conclusion holds under a more realistic paradigm in which models trained on large, out-of-domain datasets are fine-tuned on in-domain data. While pretraining and fine-tuning on in-domain data can yield strong in-domain performance—as observed by the authors—this is likely to be at the cost of general domain performance, calling into question the transferability under MDMT.

Next, we wonder if new methods might help with the issue of domain control in MDMT. The authors examine reasonable mechanisms for controlling the domain which were known at the time. Since then, new methods have been developed which we hope to investigate under the prescribed framework. We hypothesize that additive interventions (Schioppa et al., 2021), which learn tag embeddings separately from the encoder, may be harder to ignore, and that the learned interventions may be able to absorb target-side properties more easily, while freeing the encoder to learn strong representations purely for translation.

In this work we scale the original experimental

[*] Work was done during an internship at Microsoft

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 220 - 232
December 7-8, 2022 ©2022 Association for Computational Linguistics

framework presented in Pham et al. (2021) by including a significantly larger, more realistic dataset. We also experiment with additive interventions as an alternative to domain tagging. We find that:

- additive interventions perform roughly equivalently with tag-based approaches in the ideal case where provided tags match the target domain.
- additive interventions are much more robust in the face of incorrect and uncertain domain labels.
- when the experiment is scaled, models fine-tuned targeting a single domain are strong translators, but are never unmatched by other models which can service multiple domains suggesting that MDMT models in a high-resource setting are competitive with best-in-class baselines.

2 Method

As a baseline, we inject domain metadata using the tag-based approach. In this scheme, a token representing the target-side attribute, t, is prepended to source segment x and fed to the encoder E whose hidden representation is finally exposed to decoder D in a “normal" fashion:

$$\hat{y} = D(E([t] + x))$$

where $+$ indicates sequence concatenation. In tag-based approaches, the expectation is that the domain tag as a prefix acts as a conditioning variable which encourages target-side attributes to appear as desired in the final translation.

While effective and architecturally non-invasive, this method is not without downsides. Because the target token's contribution to the encoder representation is learned, there is a chance that the attribute can be ignored. To address this and other weaknesses of tag-based approaches, Schioppa et al. (2021) present the additive interventions method which requires an encoder E, a decoder D, and a separate attribute embedding layer Emb. Given a source segment x and a sentence-level attribute token t, we have

$$V = Emb(t)$$
$$\hat{y} = D(E(x) \oplus V)$$

where $\oplus$ is defined as addition broadcasted along the token dimension. Importantly, this allows prototypically discrete attributes to be represented and controlled in a *continuous* fashion, allowing for interpolation, scaling, and positionally invariant combinations, among other useful features. We note that these are somewhat analogical to an “additive" version of “source factors" approaches (Hoang et al., 2016; Sennrich and Haddow, 2016) with one major difference: additive interventions happen *after* the encoder rather than *before* the encoder.

While the original work only introduces the interventions to the top-most decoder layers in order to allow for partially freezing pretrained networks, we simplify by applying the intervention to the top layer of the encoder, such that it affects all decoder layers. Further, the authors report that improved general performance can be promoted by randomly inducing a zero-vector intervention. As such, we can specify that t is randomly replaced by $\langle\text{PAD}\rangle$ with some probability with the same effect. We report 20% masking in this paper, though we experiment with 0% masking and find no significant differences between the two.

Source	Parallel sents (k)	Source tokens (m)
ParaCrawl	229,340	4,190.0
BANK	190	6.3
IT	270	3.6
LAW	501	17.1
TALK	160	3.6
RELIG	130	3.2
MED	2,609	133.0
NEWS	254	5.6

Table 1: Effective training set sizes

3 Experimental Setup

3.1 Data

We follow the supervised data settings prescribed by Pham et al. (2021) which includes splits from seven domains of varying disparity: BANK, IT, LAW, TALK, RELIG, MED, and NEWS. These domains are drawn from various sources: the European Central Bank corpus (BANK) (Tiedemann, 2012); the documentation for the KDE, Ubuntu, GNOME, and PHP projects from Opus (Tiedemann, 2009) combined to form IT; The JRC-Acquis corpus (LAW) (Steinberger et al., 2006); TED Talks (TALK) (Cettolo et al., 2012); the Tanzil translation of the Koran (RELIG); the UFAL Medi-

Method	BANK		IT		LAW		TALK		RELIG		MED		WMT15	
	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET
`general base`	42.4	0.485	38.3	0.311	56.2	0.832	40.6	0.585	18.9	0.166	43.9	0.548	41.3	0.639
`combined base`	52.1	0.559	45.6	0.528	59.8	0.855	41.5	0.614	27.8	0.284	49.8	0.651	41.7	0.633
`combined ints`	51.9	**0.573**	44.7	0.512	59.9	0.859	41.3	0.610	27.6	0.268	50.1	0.647	**41.6**	**0.638**
`combined tags`	52.0	0.546	**46.5**	0.492	59.8	0.856	**43.7**	**0.647**	**28.8**	**0.307**	50.1	0.647	36.8	0.606
`in-dom ints`	58.5	0.615	51.9	0.615	66.6	0.891	39.2	0.494	88.7	0.872	55.4	0.695	**30.1**	**0.289**
`in-dom tags`	58.7	0.611	51.1	0.599	66.4	0.893	**39.8**	**0.531**	89.5	0.893	55.4	0.685	26.8	0.243
`multi-dom FT ints`	56.1	0.604	50.6	0.605	64.9	**0.896**	41.3	0.580	79.4	0.791	51.6	0.671	**34.3**	0.433
`multi-dom FT tags`	**56.9**	0.614	50.9	0.595	64.8	0.870	41.6	**0.605**	**83.6**	**0.850**	51.9	0.673	33.4	0.439
`single-dom FT`	58.2	0.637	50.8	0.629	67.0	0.917	45.1	0.653	39.0	0.402	52.6	0.679	–	–

Table 2: MT quality scores per test set. Statistically significant differences between `tags` and `ints` at the 95% confidence interval with 1000 bootstrapped samples **bolded**.

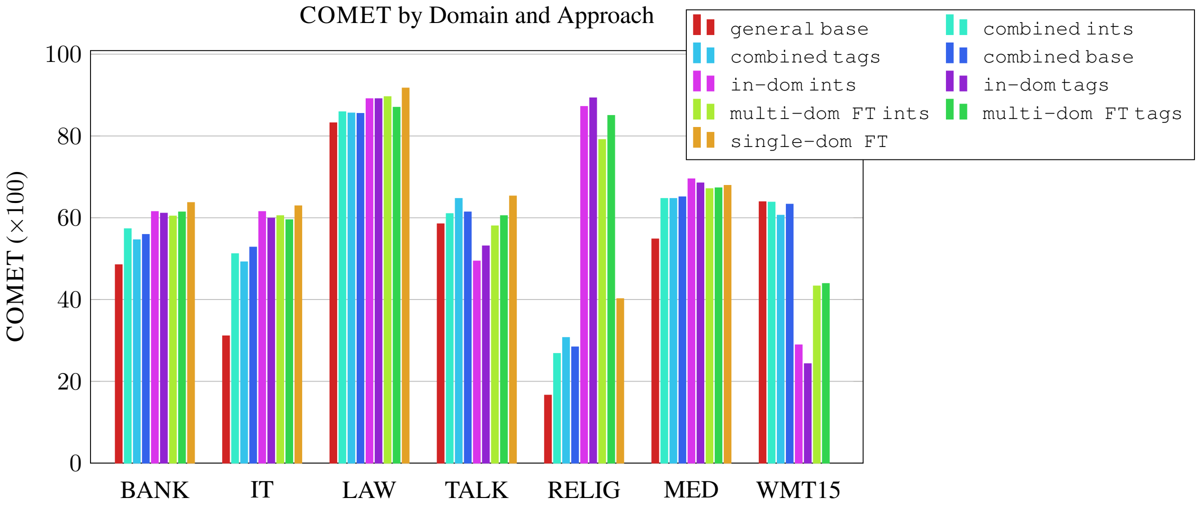


Figure 1: COMET scores (×100) by domain and approach

cal corpus v1.0 (MED)[1]; and News Commentary corpus v12 (NEWS) (Tiedemann, 2012). For sake of consistency, we rely on roughly the same splits as provided by the authors,[2], though we remove duplicates within each domain, which changes the size of each training set slightly. Additionally we include English-French ParaCrawl v9 (Bañón et al., 2020) to serve as a large out-of-domain training set for some experimental settings. The effective training set sizes are summarized in Table 1.

3.2 Models

We consider several models falling into two categories: those trained with (`control`) and without(`no control`) a method for selecting the target domain.

We use approximately the same architecture for all settings, though note that all intervention-based models have an extra embedding layer with the same embedding dimension as the encoder[3]. The basic architecture follows a 12-layer encoder, 6-layer decoder transformer with 8 attention heads each (Vaswani et al., 2017), encoder and decoder feedforward embedding dimensions of 4096, and encoder and decoder embedding dimensions of 1024.

3.2.1 `no control`

We train three models with no training-time information about the domain that the data comes from and, as a consequence, have no ability to explicitly control the target domain:

1. we have an out-of-domain baseline which is trained only on ParaCrawl: `general base`.
2. we have a model which is trained on the in-domain plus out-of-domain training sets:

[1] https://ufal.mff.cuni.cz/ufal_medical_corpus

[2] https://github.com/qmpham/experiments

[3] Adding $|D| \times 1024$ parameters, where D is the set of domain labels

combined base.

3. we have six quasi-oracle fine-tuned models which are produced by fine-tuning the general base model on each target domain's training set; we collectively refer to this set of models as single-domain fine-tuned (single-dom FT).

3.2.2 control

As mechanisms for controlling the target domain we consider:

1. prepending the domain tag to the source sequence, tags
2. additive interventions with 20% masking, ints

We apply these two methods to three settings:

1. an in-domain plus out-of-domain setting, combined
2. an in-domain-only setting, in-dom
3. a multi-domain fine-tuning setting, multi-dom FT, where general base is fine-tuned on all in-domain data with domain information available at training time.

This results in six models:

- combined ints
- in-dom ints
- multi-dom FT ints
- combined tags
- in-dom tags
- multi-dom FT tags.

3.3 Training

We train a joint unigram segmentation model (Kudo, 2018) using SentencePiece (Kudo and Richardson, 2018) with a vocabulary of size 32k for each setting in general base, combined, and in-dom (reusing general base's model for multi-dom FT and single-dom FT). We train each model by sampling 10M sentences randomly, splitting on digits and enabling byte-fallback. We add a special token for each domain for which we have splits: ⟨BANK⟩, ⟨IT⟩, ⟨LAW⟩, ⟨TALK⟩, ⟨RELIG⟩, ⟨MED⟩, and ⟨NEWS⟩. We use these models to segment the data as appropriate in each setting.

We use dropout of 0.1 but disable attention dropout and ReLU dropout. We optimize label smoothed cross-entropy loss with a label smoothing factor of 0.1 (Szegedy et al., 2016) using Adam (Kingma and Ba, 2015). All models are built and trained using fairseq (Ott et al., 2019).

For models trained with out-of-domain data, we shard the effective dataset with each shard containing approximately 1b target tokens. For models trained with in-domain data only, we consider the entire combined in-domain dataset to be a single shard. We train for 30 virtual epochs, where a virtual epoch is defined as a single pass over one shard. For models which are fine-tuned, we fine-tune for 10 additional virtual epochs.

Each in-domain training set is assigned a unique special token which is included in the vocabulary and examples drawn from these in-domain training sets are provided the associated special token at training time. Examples from ParaCrawl are assigned no special domain token (i.e., no token is prepended in tags models and ⟨PAD⟩ is always provided in ints models).

3.4 Evaluation

We evaluate in three settings to probe various aspects of MT quality:

- we evaluate in-domain performance with each model from control and no control to determine the relative effectiveness of the methods of control against methods without control.
- we evaluate on the WMT15 English-French test set (Bojar et al., 2015) with no domain label provided (i.e., as if the models were in the no control setting) to test catastrophic forgetting (Goodfellow et al., 2013) in a general setting. Importantly, while the models trained on in-domain data have been exposed to newswire data, the labels are not provided at test time in this setting.
- we evaluate the effect of providing the incorrect tag to each test set, as computed by SacreBLEU (Post, 2018) and COMET (Rei et al., 2020), to test the resilience of models to label errors

4 Results

No clear winner in ideal case We evaluate the setting in which the provided domain label matches

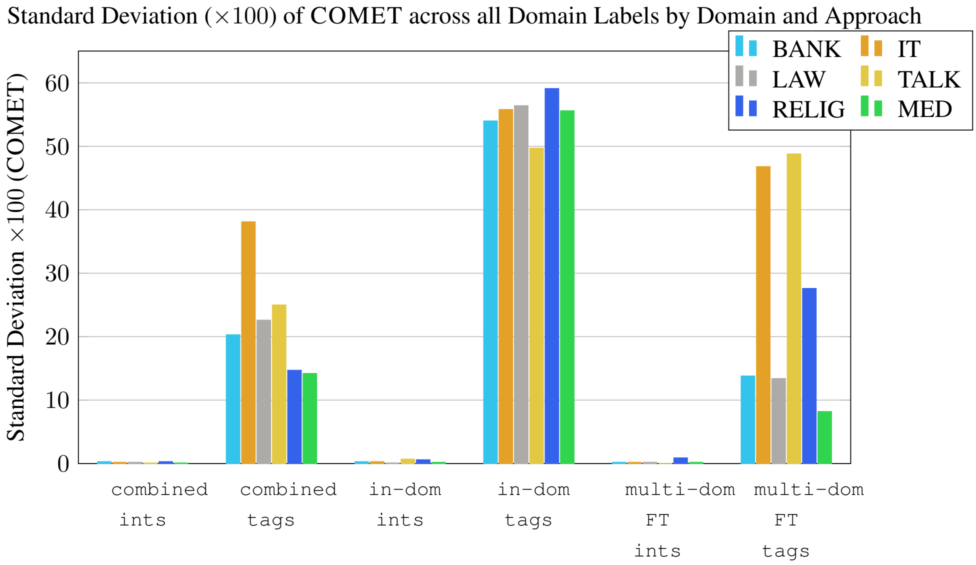
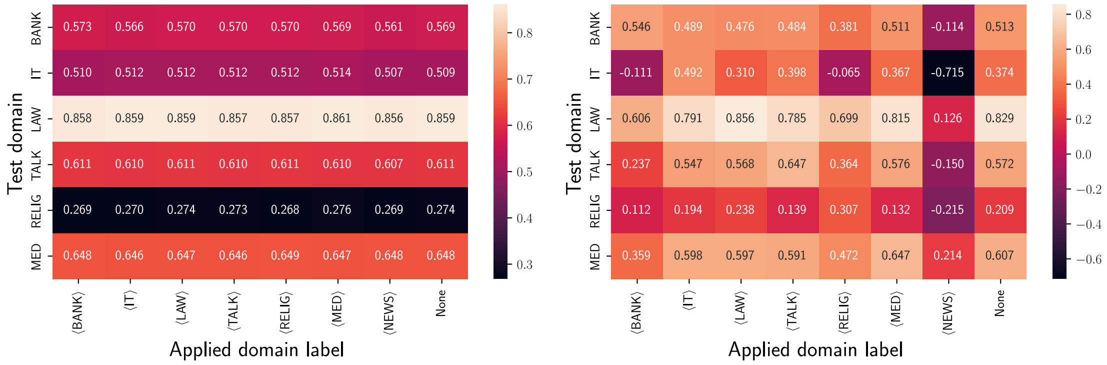


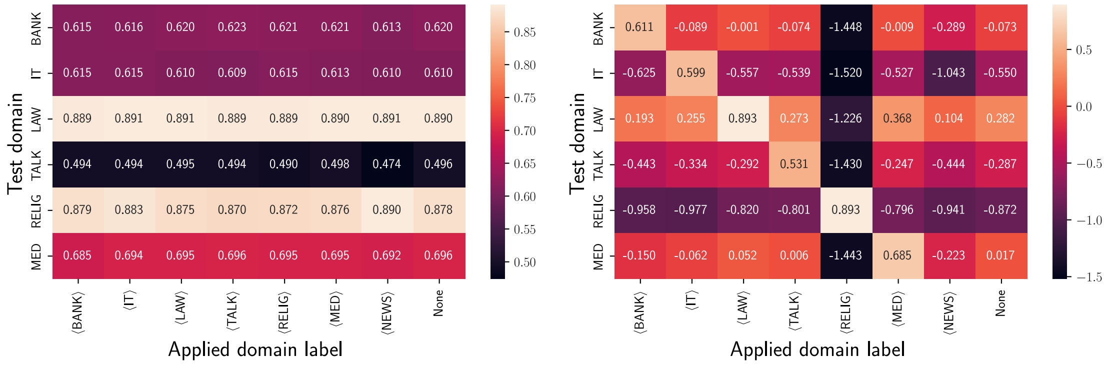
Figure 2: Impact of domain label error on COMET per test set and approach



	⟨BANK⟩	⟨IT⟩	⟨LAW⟩	⟨TALK⟩	⟨RELIG⟩	⟨MED⟩	⟨NEWS⟩	None
BANK	0.573	0.566	0.570	0.570	0.570	0.569	0.561	0.569
IT	0.510	0.512	0.512	0.512	0.512	0.514	0.507	0.509
LAW	0.858	0.859	0.859	0.857	0.857	0.861	0.856	0.859
TALK	0.611	0.610	0.611	0.610	0.611	0.610	0.607	0.611
RELIG	0.269	0.270	0.274	0.273	0.268	0.276	0.269	0.274
MED	0.648	0.646	0.647	0.646	0.649	0.647	0.648	0.648

	⟨BANK⟩	⟨IT⟩	⟨LAW⟩	⟨TALK⟩	⟨RELIG⟩	⟨MED⟩	⟨NEWS⟩	None
BANK	0.546	0.489	0.476	0.484	0.381	0.511	-0.114	0.513
IT	-0.111	0.492	0.310	0.398	-0.065	0.367	-0.715	0.374
LAW	0.606	0.791	0.856	0.785	0.699	0.815	0.126	0.829
TALK	0.237	0.547	0.568	0.647	0.364	0.576	-0.150	0.572
RELIG	0.112	0.194	0.238	0.139	0.307	0.132	-0.215	0.209
MED	0.359	0.598	0.597	0.591	0.472	0.647	0.214	0.607

Figure 3: COMET of combined models under various domain labels. ints left, tags right. ints maintain high quality translations under mismatching domain labels in all cases, unlike tags.



	⟨BANK⟩	⟨IT⟩	⟨LAW⟩	⟨TALK⟩	⟨RELIG⟩	⟨MED⟩	⟨NEWS⟩	None
BANK	0.615	0.616	0.620	0.623	0.621	0.621	0.613	0.620
IT	0.615	0.615	0.610	0.609	0.615	0.613	0.610	0.610
LAW	0.889	0.891	0.891	0.889	0.889	0.890	0.891	0.890
TALK	0.494	0.494	0.495	0.494	0.490	0.498	0.474	0.496
RELIG	0.879	0.883	0.875	0.870	0.872	0.876	0.890	0.878
MED	0.685	0.694	0.695	0.696	0.695	0.695	0.692	0.696

	⟨BANK⟩	⟨IT⟩	⟨LAW⟩	⟨TALK⟩	⟨RELIG⟩	⟨MED⟩	⟨NEWS⟩	None
BANK	0.611	-0.089	-0.001	-0.074	-1.448	-0.009	-0.289	-0.073
IT	-0.625	0.599	-0.557	-0.539	-1.520	-0.527	-1.043	-0.550
LAW	0.193	0.255	0.893	0.273	-1.226	0.368	0.104	0.282
TALK	-0.443	-0.334	-0.292	0.531	-1.430	-0.247	-0.444	-0.287
RELIG	-0.958	-0.977	-0.820	-0.801	0.893	-0.796	-0.941	-0.872
MED	-0.150	-0.062	0.052	0.006	-1.443	0.685	-0.223	0.017

Figure 4: COMET of in-dom models under various domain labels. ints left, tags right. ints maintain high quality translations under mismatching domain labels in all cases, unlike tags.

the target test domain, and the setting of WMT15 without a provided domain label, for each setting apart from single-dom FT. The results can be read in Table 2 and are visualized in Figure 1.

Table 2 shows that when comparing control models within a training setting using bootstrap resampling (sample sizes of 1000) (Koehn, 2004), the difference in performance of tags and ints are insignificant in the majority of cases. While there are a few cases of statistically significant differ-

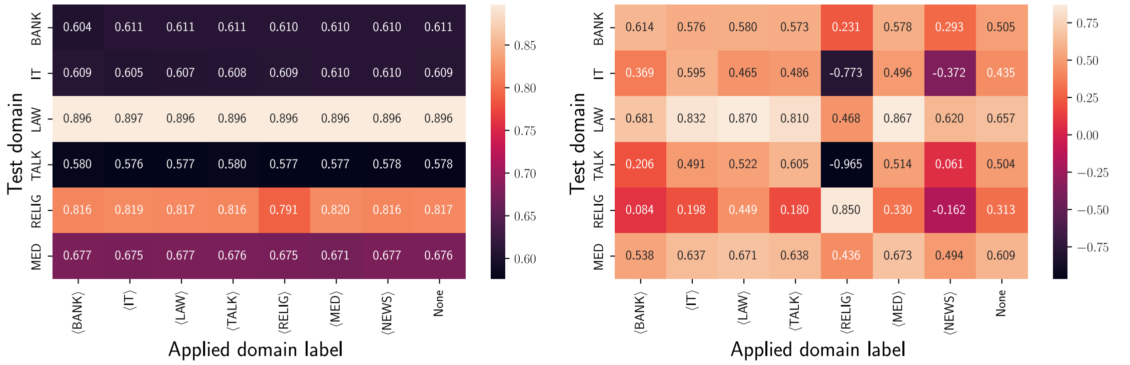


Figure 5: COMET of `multi-dom FT` models under various domain labels. `ints` left, `tags` right. `ints` maintain high quality translations under mismatching domain labels in all cases, unlike `tags`.

ences, neither `tags` nor `ints` are uniformly preferred in these cases. The opposite is observed on the out-of-domain WMT15, where `ints` performs uniformly better than `tags`, often significantly.

We observe that methods with `control` in the `combined` setting perform approximately equally to the `combined base`, showing that naive combination of in-domain and out-of-domain with a mechanism to control the domain does not improve over approaches without control, though `in-dom` and `multi-dom FT` models tend to perform better on average than any model in the `combined` setting.

`ints` are robust under domain label mismatch Next, we perform an ablation study in which we score each test across all domain label assignments (including the correct label and no label), which allows us to observe the effects of test-time labeling error. While we compute both BLEU and COMET, we include only COMET here.[4] We include the full results in Tables 3–8, but summarize the findings in Figures 2-5, which show the robustness of various models and settings to mislabeled domains.

Figures 3–5 show heatmaps resulting from this ablation, but we refer interested readers to Tables 3–8 for the long-form charts. We see that `tags` systems' performances vary dramatically, incurring severe degradation in the face of domain label error but performing strongest along the diagonal. `ints` systems, on the other hand, see only small performance changes when provided with incorrect domain labels and roughly equal performance under all possible labels, as observed in Figure 2. We see that `in-dom tags` have the highest average variation in performance, likely owing to the small amount of data which suggests that `in-dom tags` overfits to the training data. The variation in performance of `ints` systems approaches that of the `general base`, which by definition ignores the domain label and therefore has 0 variance; however, `ints` has demonstrably stronger performance than `general base` in all domains and, indeed, stronger performance than `tags` in a handful of domains and thus seems to learn strong general representations for translation which disentangles the representations of the encoder from the representations of the attribute.

Additionally, through manual analysis we find that `tags` systems are more prone to hallucinating translation artifacts from the corpus associated with the domain label being used, often causing quality degradation. We refer to Table 15 for an example of such artifacts, which includes topical and target language mismatches along with tokens which appear as a result of the HTML-encoded nature of the ⟨IT⟩ dataset.[5]

Single-domain fine-tuning is not as competitive in large-data settings We compare the performance of models trained only with in-domain data and out-of-domain data. From Table 2, we see slightly stronger in-domain performance for `in-dom` models as compared to models fine-tuned with out-of-domain data at the cost of out-of-domain performance on WMT15, suggesting that `multi-dom FT` models generalize better and may surpass `in-dom` models with more training due to the relatively little fine-tuning budget of 10 epochs afforded to them comparatively.

[4] Similar results for BLEU are listed in Appendix A.2

[5] Escaping seems to be an artifact of Moses preprocessing leakage of raw data; not germane to all domains in this work.

Finally, we see that while `single-dom FT` is typically among the highest performing systems for a given test set, it is never unmatched by an alternative system in `control`. We observe that `single-dom FT` is uniformly stronger than `general base` and `combined`, `in-dom` and `multi-dom FT` show competitive in-domain performance. We note that because there is one `single-dom FT` model per test set, the effective parameter budget is six times larger than any of the individual models, providing support for both its impracticality and untenability as compared to any other setting. This suggests that single-domain fine-tuning is not as effective as expected in high-resource settings as a strong upper-bound in MDMT.

5 Related Work

Incorporating extra-sentential information has a rich history in NMT. Aside from controlling for the domain, Sennrich et al. (2016) use a politeness tag at training and inference time to accommodate coarse politeness control in machine translation. Additionally, Kuczmarski and Johnson (2018) use tags to afford users the ability to vary binary gender in the translations of gender-neutral inputs, hoping to address gender bias in MT.

At the sub-sequence level, Hoang et al. (2016) and Sennrich and Haddow (2016) included linguistically-informed word-level "source factors", such as part-of-speech tags and dependency relations, as additional feature factors to be concatenated to form a full encoder representation with the goal of reducing ambiguity and sparseness issues.

Perhaps more relatedly, several works have explored the impacts of incorporating domain information into training using various methods. Kobus et al. (2017) explore two methods: a tag-based approach which concatenates a special token to the end of the source sequence, and a "source factors"-style approach which concatenates domain-level embeddings to each token embedding in the source. Sharaf et al. (2020) explore few-shot domain adaptation, rather than domain control, through the lens of meta-learning and show that a meta-learning based approach is generally stronger than other adaptation approaches, though we note that adaptation and control address different needs. Finally, Stojanovski and Fraser (2021) frame machine translation with document-context as an unsupervised domain adaptation problem and incorporate domain embeddings within the encoder, summed with positional and word embeddings, yielding strong improvements over competitive baseline models.

6 Conclusion

In this work we examined the relative impact of additive interventions in a large-scale MDMT setting. We find that typically there are no significant differences between additive interventions and tag-based approaches when the provided domain label matches the test set, but find that additive interventions exhibit *much more desirable degradation properties* when the domain label is unknown or incorrectly provided. In addition, we find that models first trained on a large, general corpus and then fine-tuned on a single-domain—a realistic baseline in machine translation—rarely perform significantly better than approaches which are trained or fine-tuned only on in-domain data, which is in contrast to their generally superior performance in low-resource settings.

In future work we consider developing extensions to additive interventions which can further improve their performance in MDMT settings. Additionally, studying additive interventions in other tasks where tag-based approaches are dominant, such as multi-lingual machine translation, could be an interesting avenue for exploration.

References

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual* *conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy. European Association for Machine Translation.

Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks.

Cong Duy Vu Hoang, Gholamreza Haffari, and Trevor Cohn. 2016. Improving neural translation models with linguistic factors. In *Proceedings of the Australasian Language Technology Association Workshop 2016*, pages 7–14, Melbourne, Australia.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378, Varna, Bulgaria. INCOMA Ltd.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

James Kuczmarski and Melvin Johnson. 2018. Gender-aware natural language translation. *Technical Disclosure Commons, (October 08, 2018)*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

MinhQuang Pham, Josep Maria Crego, and François Yvon. 2021. Revisiting multi-domain machine translation. *Transactions of the Association for Computational Linguistics*, 9:17–35.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Andrea Schioppa, David Vilar, Artem Sokolov, and Katja Filippova. 2021. Controlling machine translation for multiple attributes with additive interventions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6676–6696, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40, San Diego, California. Association for Computational Linguistics.

Amr Sharaf, Hany Hassan, and Hal Daumé III. 2020. Meta-learning for few-shot NMT adaptation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 43–53, Online. Association for Computational Linguistics.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and Dániel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Dario Stojanovski and Alexander Fraser. 2021. Addressing zero-resource domains using document-level context in neural machine translation. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 80–93, Kyiv, Ukraine. Association for Computational Linguistics.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Jörg Tiedemann. 2009. News from opus — a collection of multilingual parallel corpora with tools and interfaces. *Advances in Natural Language Processing*, pages 237–248.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

A Raw scores

A.1 Ablation (COMET)

Provided label Test set	⟨BANK⟩	⟨IT⟩	⟨LAW⟩	⟨TALK⟩	⟨RELIG⟩	⟨MED⟩	⟨NEWS⟩	None
BANK	0.573	0.566	0.570	0.570	0.570	0.569	0.561	0.569
IT	0.510	0.512	0.512	0.512	0.512	0.514	0.507	0.509
LAW	0.858	0.859	0.859	0.857	0.857	0.861	0.856	0.859
TALK	0.611	0.610	0.611	0.610	0.611	0.610	0.607	0.611
RELIG	0.269	0.270	0.274	0.273	0.268	0.276	0.269	0.274
MED	0.648	0.646	0.647	0.646	0.649	0.647	0.648	0.648

Table 3: COMET scores of `combined ints` under various domain labels

Provided label Test set	⟨BANK⟩	⟨IT⟩	⟨LAW⟩	⟨TALK⟩	⟨RELIG⟩	⟨MED⟩	⟨NEWS⟩	None
BANK	0.546	0.489	0.476	0.484	0.381	0.511	-0.114	0.513
IT	-0.111	0.492	0.310	0.398	-0.065	0.367	-0.715	0.374
LAW	0.606	0.791	0.856	0.785	0.699	0.815	0.126	0.829
TALK	0.237	0.547	0.568	0.647	0.364	0.576	-0.150	0.572
RELIG	0.112	0.194	0.238	0.139	0.307	0.132	-0.215	0.209
MED	0.359	0.598	0.597	0.591	0.472	0.647	0.214	0.607

Table 4: COMET scores of `combined tags` under various domain labels

Provided label Test set	⟨BANK⟩	⟨IT⟩	⟨LAW⟩	⟨TALK⟩	⟨RELIG⟩	⟨MED⟩	⟨NEWS⟩	None
BANK	0.615	0.616	0.620	0.623	0.621	0.621	0.613	0.620
IT	0.615	0.615	0.610	0.609	0.615	0.613	0.610	0.610
LAW	0.889	0.891	0.891	0.889	0.889	0.890	0.891	0.890
TALK	0.494	0.494	0.495	0.494	0.490	0.498	0.474	0.496
RELIG	0.879	0.883	0.875	0.870	0.872	0.876	0.890	0.878
MED	0.685	0.694	0.695	0.696	0.695	0.695	0.692	0.696

Table 5: COMET scores of `in-dom ints` under various domain labels

Provided label Test set	⟨BANK⟩	⟨IT⟩	⟨LAW⟩	⟨TALK⟩	⟨RELIG⟩	⟨MED⟩	⟨NEWS⟩	None
BANK	0.611	-0.089	-0.001	-0.074	-1.448	-0.009	-0.289	-0.073
IT	-0.625	0.599	-0.557	-0.539	-1.520	-0.527	-1.043	-0.550
LAW	0.193	0.255	0.893	0.273	-1.226	0.368	0.104	0.282
TALK	-0.443	-0.334	-0.292	0.531	-1.430	-0.247	-0.444	-0.287
RELIG	-0.958	-0.977	-0.820	-0.801	0.893	-0.796	-0.941	-0.872
MED	-0.150	-0.062	0.052	0.006	-1.443	0.685	-0.223	0.017

Table 6: COMET scores of `in-dom tags` under various domain labels

Provided label Test set	⟨BANK⟩	⟨IT⟩	⟨LAW⟩	⟨TALK⟩	⟨RELIG⟩	⟨MED⟩	⟨NEWS⟩	None
BANK	0.604	0.611	0.611	0.611	0.610	0.610	0.610	0.611
IT	0.609	0.605	0.607	0.608	0.609	0.610	0.610	0.609
LAW	0.896	0.897	0.896	0.896	0.896	0.896	0.896	0.896
TALK	0.580	0.576	0.577	0.580	0.577	0.577	0.578	0.578
RELIG	0.816	0.819	0.817	0.816	0.791	0.820	0.816	0.817
MED	0.677	0.675	0.677	0.676	0.675	0.671	0.677	0.676

Table 7: COMET scores of `multi-dom FT ints` under various domain labels

Provided label Test set	⟨BANK⟩	⟨IT⟩	⟨LAW⟩	⟨TALK⟩	⟨RELIG⟩	⟨MED⟩	⟨NEWS⟩	None
BANK	0.614	0.576	0.580	0.573	0.231	0.578	0.293	0.505
IT	0.369	0.595	0.465	0.486	-0.773	0.496	-0.372	0.435
LAW	0.681	0.832	0.870	0.810	0.468	0.867	0.620	0.657
TALK	0.206	0.491	0.522	0.605	-0.965	0.514	0.061	0.504
RELIG	0.084	0.198	0.449	0.180	0.850	0.330	-0.162	0.313
MED	0.538	0.637	0.671	0.638	0.436	0.673	0.494	0.609

Table 8: COMET scores of `multi-dom FT tags` under various domain labels

A.2 Ablation (BLEU)

All scores reported are from SacreBLEU[6] (Post, 2018).

Provided label Test set	⟨BANK⟩	⟨IT⟩	⟨LAW⟩	⟨TALK⟩	⟨RELIG⟩	⟨MED⟩	⟨NEWS⟩	None
BANK	51.9	51.7	51.9	51.9	51.9	51.8	51.8	51.9
IT	44.6	44.7	44.8	44.8	44.6	44.7	44.7	44.6
LAW	59.8	59.8	59.9	59.8	59.7	59.8	59.7	59.9
TALK	41.3	41.3	41.4	41.3	41.4	41.3	41.1	41.5
RELIG	27.6	27.8	27.7	27.8	27.6	27.9	27.5	27.7
MED	50.0	50.0	50.0	50.0	49.9	50.1	50.0	50.0

Table 9: BLEU scores of `combined ints` under various domain labels

Provided label Test set	⟨BANK⟩	⟨IT⟩	⟨LAW⟩	⟨TALK⟩	⟨RELIG⟩	⟨MED⟩	⟨NEWS⟩	None
BANK	52.0	43.5	43.0	40.4	39.0	45.2	30.2	44.2
IT	18.5	46.5	36.3	39.9	26.5	37.2	11.0	35.0
LAW	50.2	56.4	59.8	50.7	51.4	55.5	36.9	56.2
TALK	29.5	39.2	38.1	43.7	28.3	39.7	22.7	37.1
RELIG	21.6	24.4	25.5	16.3	28.8	18.9	14.5	22.6
MED	43.5	48.5	48.3	47.3	45.0	50.1	41.6	49.1

Table 10: BLEU scores of `combined tags` under various domain labels

Provided label Test set	⟨BANK⟩	⟨IT⟩	⟨LAW⟩	⟨TALK⟩	⟨RELIG⟩	⟨MED⟩	⟨NEWS⟩	None
BANK	58.5	58.6	58.6	58.6	58.5	58.8	58.2	58.7
IT	52.0	51.9	51.4	51.4	51.8	51.6	51.4	51.8
LAW	66.1	66.2	66.1	66.0	65.9	66.1	66.0	66.1
TALK	39.0	39.1	39.1	39.2	39.1	39.2	38.8	39.0
RELIG	89.2	89.2	89.0	88.7	88.7	89.2	89.3	89.1
MED	55.4	55.5	55.3	55.4	55.4	55.4	55.4	55.5

Table 11: BLEU scores of `in-dom ints` under various domain labels

Provided label Test set	⟨BANK⟩	⟨IT⟩	⟨LAW⟩	⟨TALK⟩	⟨RELIG⟩	⟨MED⟩	⟨NEWS⟩	None
BANK	58.7	31.2	36.0	34.3	3.9	36.1	27.3	34.4
IT	15.5	51.1	16.6	20.0	0.4	18.8	5.9	15.9
LAW	42.2	43.5	66.4	45.3	12.4	48.2	40.2	44.7
TALK	18.6	21.0	20.7	39.8	1.0	23.8	17.2	21.5
RELIG	6.2	6.1	8.2	8.7	89.5	9.0	5.5	7.6
MED	32.2	33.2	32.8	33.3	5.5	55.4	29.5	33.5

Table 12: BLEU scores of `in-dom tags` under various domain labels

[6] `BLEU|nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.2.0`

Provided label Test set	⟨BANK⟩	⟨IT⟩	⟨LAW⟩	⟨TALK⟩	⟨RELIG⟩	⟨MED⟩	⟨NEWS⟩	None
BANK	56.1	55.9	56.5	56	56.1	56.4	55.3	56.3
IT	50.6	50.6	50.0	50.4	50.3	50.6	49.8	50.9
LAW	64.8	64.7	64.9	64.9	64.8	65.2	64.5	65.0
TALK	41.2	40.8	41.1	41.3	41.3	41.2	40.4	41.5
RELIG	80.4	81.1	80.5	80.2	79.4	81.8	79.3	82.2
MED	51.7	51.3	51.6	51.7	51.7	51.6	51.3	51.7

Table 13: BLEU scores of `multi-dom FT ints` under various domain labels

Provided label Test set	⟨BANK⟩	⟨IT⟩	⟨LAW⟩	⟨TALK⟩	⟨RELIG⟩	⟨MED⟩	⟨NEWS⟩	None
BANK	56.9	54.5	54.4	52.0	49.6	55.0	43.4	54.9
IT	43.1	50.9	47.4	46.9	28.0	46.9	17.3	40.8
LAW	55.7	63.7	64.8	61.2	59.4	64.2	55.9	60.3
TALK	28.0	37.4	36.1	41.6	8.4	36.1	23.1	36.2
RELIG	32.6	38.6	61.9	22.9	83.6	50.7	19.2	49.1
MED	49.6	51.4	51.8	50.4	49.4	51.9	49.7	51.2

Table 14: BLEU scores of `multi-dom FT tags` under various domain labels

B Figures (BLEU)

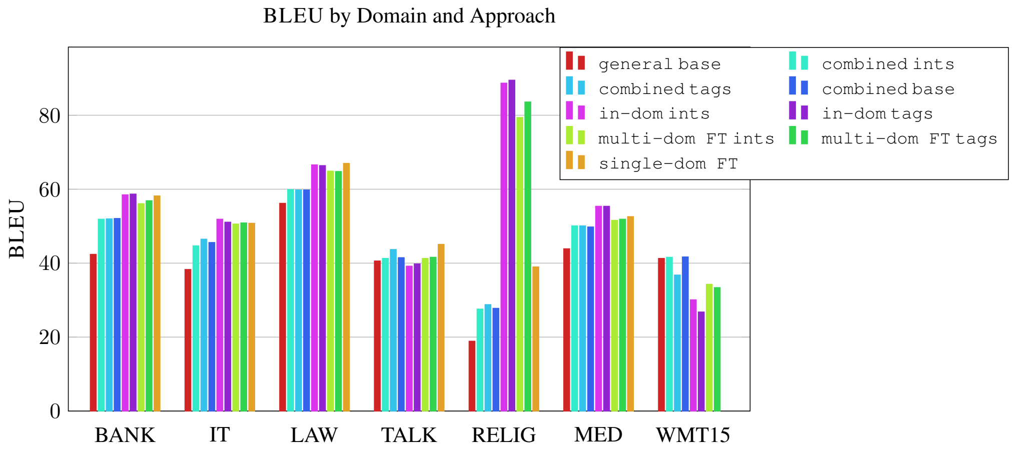


Figure 6: BLEU scores by domain and approach

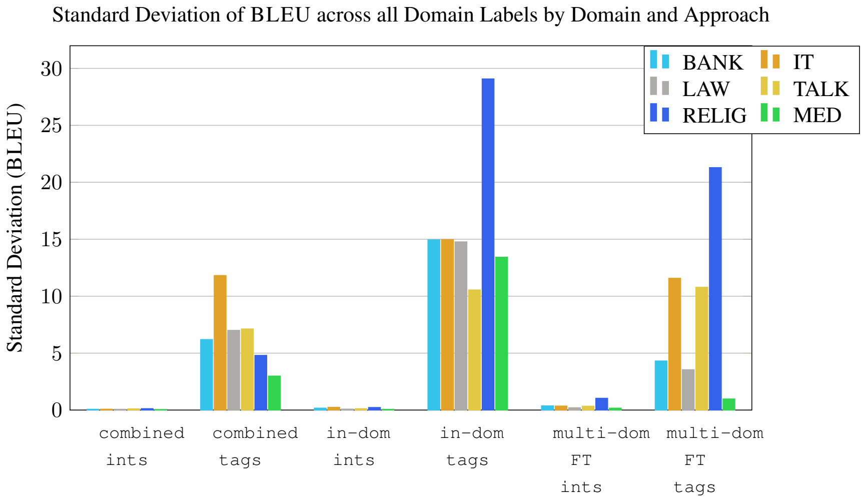


Figure 7: Impact of domain label error on BLEU per test set and approach

C Examples

Src	Never; soon they will deny ever worshipping them, and will turn into their opponents.
Ref	Bien au contraire! [ces divinités] renieront leur adoration et seront pour eux des adversaires.
`multi-dom FT ints`	Bien au contraire! [ces divinités] renieront leur adoration et seront pour eux des adversaires.
`multi-dom FT tags`	You are about to translate the ’None ’COMMAND, there are some rules on how to translate it. Please see http: / / / /www.mysql.com /.
Src	And the evil-doers say: Ye are but following a man bewitched.
Ref	Les injustes disent: «Vous ne suivez qu’un homme ensorcelé».
`in-dom ints`	Les injustes disent: «Vous ne suivez qu’un homme ensorcelé».
`in-dom tags`	Et les « & #160; diaboliques & #160; » disent & #160;: « & #160; fired & #160; » est le suivant d’un homme.

Table 15: Example translation artifacts from incorrect domain label; a translation of ⟨RELIG⟩ sentences with ⟨IT⟩ domain label under different models. We note that the HTML-encoded artifact “& #160;" appears with high frequency in ⟨IT⟩.

Inria-ALMAnaCH at the WMT 2022 shared task: Does Transcription Help Cross-Script Machine Translation?

Jesujoba O. Alabi[1]* Lydia Nishimwe[2] Benjamin Muller[2,3]
Camille Rey[2] Benoît Sagot[2] Rachel Bawden[2]
[1]Spoken Language Systems (LSV), Saarland University, Saarland Informatics Campus, Germany
[2]Inria, Paris, France [3]Sorbonne Université, France
jalabi@lsv.uni-saarland.de firstname.lastname@inria.fr

Abstract

This paper describes the Inria ALMAnaCH team submission to the WMT 2022 general translation shared task. Participating in the language directions {cs,ru,uk}→en and cs↔uk, we experiment with the use of a dedicated Latin-script transcription convention aimed at representing all Slavic languages involved in a way that maximises character- and word-level correspondences between them as well as with the English language. Our hypothesis was that bringing the source and target language closer could have a positive impact on machine translation results. We provide multiple comparisons, including bilingual and multilingual baselines, with and without transcription. Initial results indicate that the transcription strategy was not successful, resulting in lower results than baselines. We nevertheless submitted our multilingual, transcribed models as our primary systems, and in this paper provide some indications as to why we got these negative results.

1 Introduction

This paper describes the Inria ALMAnaCH team submission to the WMT 2022 general translation shared task. We chose to explore the language directions {cs,ru,uk}↔en and cs↔uk in order to concentrate on the Slavic language family. Due to some experimental problems that impacted the into-Slavic directions most heavily, we only submitted {cs,ru,uk}→en and cs↔uk language directions, but we present all results we obtained here.

A major area of interest in machine translation (MT) research is transfer between languages, particularly related ones and for lesser resourced languages (Zoph et al., 2016; Kocmi and Bojar, 2018). One way of encouraging transfer is to train multilingual models, whereby several language directions are trained simultaneously, often sharing some (Firat et al., 2016) or all model parameters (Ha et al., 2016; Johnson et al., 2017; Aharoni et al., 2019), with the hope that similarities between the languages can boost performance, particularly for the lower-resourced languages.

To encourage lexical sharing and therefore the transfer capacity of such models, joint subword segmentation models (Sennrich et al., 2016b) and MT vocabularies are often used (Sennrich et al., 2016a), and techniques such as phonetisation and transliteration/transcription can be applied to texts in a bid to overcome differences in writing systems and spelling (Nguyen and Chiang, 2017; Chakravarthi et al., 2019; Goyal et al., 2020; Muller et al., 2021).

In our submission to the WMT 2022 general translation shared task, we experimented with multilingual models and the use of customised transcription into a common writing system designed to maximise lexical sharing, similar to the one used in (Muller et al., 2021). We choose to work with the language directions involving Slavic languages, that is {cs,ru,uk}↔en and cs↔uk. We find that our transcription method unfortunately leads to degraded results, likely a consequence of errors being injected and notably the necessity to apply a learned detranscription model as a post-processing step for into-Slavic language directions. Our multilingual models achieved largely inferior results to our bilingual baseline models for the same number of parameters, showing that multilingual transfer cannot compensated for sharing the vocabulary over a larger number of languages. Transcribing the languages in the multilingual setup results narrows the gap slightly, but the results remain lower than the bilingual baselines. We nevertheless decided to submit our multilingual models with common-Slavic transcription rather than our superior baseline results in the full knowledge that these results would not achieve the best results in the shared task.[1]

[*] Contributions made whilst at Inria.

[1] We believe it was more interesting to submit these results to test our hypothesis rather than to submit more standard

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 233 - 243
December 7-8, 2022 ©2022 Association for Computational Linguistics

cs	original	*Sníh pokryl stromy vedle zámku.*
cs	transcribed	*Snig pokril stromi vedle zamku.*
uk	original	Сніг вкрив дерева біля замку.
uk	transliterated	*Snih vkryv dereva bilja zamku.*
uk	transcribed	*Sneg vkriv dereva bela zamku.*
ru	original	Снег покрыл деревья возле замка.
ru	transliterated	*Sneg pokryl derev'ja vozle zamka.*
ru	transcribed	*Sneg pokril dereva vozle zamka.*
en	original	The snow has covered the trees next to the castle.

Table 1: Constructed example illustrating the difference between standard transliteration and our linguistically motivated transcription.

2 Related Work

There has been a considerable body of work in MT dedicated to multilingual models, whereby several language directions are trained simultaneously, with different degrees of parameter sharing, ranging from separate encoders and decoders (Firat et al., 2016) to the sharing of a single encoder and a single decoder for all languages with a single shared vocabulary (Ha et al., 2016; Johnson et al., 2017; Aharoni et al., 2019). As well as being practical by providing a single MT model that can be used for multiple directions, the models have the advantage of aiding the representations of lower-resourced languages, particularly if related, higher-resourced languages are also included in training (Kudugunta et al., 2019; Aharoni et al., 2019; Tchistiakova et al., 2021).

In addition to approaches such as joint subword segmentation models (Sennrich et al., 2016b) and the use of a joint vocabulary for all languages (Johnson et al., 2017), strategies to encourage more lexical sharing have also been explored in order to overcome surface differences introduced by orthographic conventions, notably phonetisation (Liu et al., 2019; Rosales Núñez et al., 2019; Sun et al., 2022) and transliteration (Nakov and Tiedemann, 2012; Nguyen and Chiang, 2017; Goyal et al., 2020). These approaches can be particularly useful for borrowings and for proper nouns, which can be made to be identical (or near-identical) across languages once transliteration has been applied.

Transliteration is the mapping of one writing system to another, and therefore is relevant when languages are written in different scripts (e.g. Latin, Cyrillic, Devanagari, etc.). In particular for related languages, it can be interesting to apply transliteration in order to exploit the fact that many words can be made to be similar on the surface once transliteration has been applied. Much of the work that has explored transliteration for MT has focused on Indian languages, for which the mapping between scripts is relatively straightforward (Bawden et al., 2019; Goyal et al., 2020; Kunchukuttan and Bhattacharyya, 2021; Sun et al., 2022), but there has also been research on other language families (Maimaiti et al., 2019; Sun et al., 2022), including Slavic languages (Maimaiti et al., 2019). In our systems, we follow a similar approach to test whether a form of transliteration that maximises lexical overlap between Slavic languages could help translation in a multilingual setup, even in the relatively high-resource scenario provided by the shared task.

3 Multilingual Slavic models with transcription

Building on the previous work on multilingual MT and on transliteration to encourage lexical sharing, we propose multilingual models with a custom linguistically motivated transcription scheme for translation between English and the Slavic languages Czech (cs), Ukrainian (uk) and Russian (ru).

Multilingual Slavic translation models We train multilingual Slavic translation models with a single encoder-decoder architecture as in (Johnson et al., 2017) over the following language directions: {cs,uk,ru} from and into English and cs to and from uk. Given that a single shared encoder and a single shared decoder is used, the same vocabulary is used across all languages, and we also share embeddings across the encoder and decoder. To further encourage sharing, we train a joint subword segmentation model. To test the performance of this multilingual model, we compare against bilingual baselines trained uniquely on parallel data for the

baseline systems. Due to human error, these submitted models perform less well than the results presented in this paper, as described in Section 5.3.

specific language pair, which also share encoder and decoding embeddings.

Linguistically motivated transcription We experiment with the use of a customised common Slavic writing system designed with the aim of maximising lexical overlap between the Slavic languages we study. The underlying idea is that MT models, both bilingual and multilingual, should benefit from an increase in the similarity between languages including in training. Since Slavic languages share a common ancestor, Proto-Slavic, they display similarities in terms of phonetics, grammar and vocabulary. Lexical overlap, though, can be further improved in at least two ways:

- Whereas Czech uses the Latin script with a number of diacritics, Russian and Ukrainian use the Cyrillic script. Using a common script would inevitably increase the lexical overlap and make it more explicit. For instance, using a standard Latin transliteration scheme for Russian,[2] the Russian word рука 'hand' can be rendered as *ruka*, which is identical to Czech *ruka* 'hand'.

- Each Slavic language has undergone a number of changes from Proto-Slavic, including regular sound changes. Examples such as Ru. рука∼*ruka* vs. Cz. *ruka*, where transliteration alone is enough to create a perfect lexical overlap, are therefore rare. However, there are a large number of cognates (words in related languages that share a common ancestor), which, independently of the script, are still similar and only differ in partly systematic ways. For instance, Ru. корень 'root', Uk. корiнь 'id.' and Cz. *kořen* 'id.' are cognates. Using standard transliteration schemes, the Russian and Ukrainian words can be rendered as *koren'* and *korin'*, respectively. This is closer to Cz. *kořen* but is not identical. More importantly, it fails to identify the fact that Uk. i often corresponds to Cz. *e* and that Cz. *ř* often corresponds to Ru. and Uk. р.

To further increase lexical overlap and with the aim of encouraging more transfer between the languages than what is permitted by standard transliteration schemes, we developed transformation rules for all three Slavic languages based on systematic patterns, based on observations from cognate lists in the three languages and knowledge about their morphology, in order to lower the differences introduced between them by sound changes and morphological particularities, similarly to (Muller et al., 2021). For Russian and Ukrainian, this involves a script change, but Czech is also modified. We call this transformation *linguistically motivated transcription*.[3] Going back to the example above, the output of our transcription scripts for Ru. корень, Uk. корiнь and Cz. *kořen* is the same, namely *koren*. Table 1 illustrates our linguistically motivated transcription strategies on a constructed multilingual example.

Transcription and detranscription Our common Slavic transcription is applied during preprocessing to the training data. For into-English language directions, no further processing is required following translation, because we only transcribe the Slavic languages and not English. However for from-English directions and for cs↔uk, the output of the MT model will require detranscription in order to transform the outputs into the correct form for that language. We therefore also train small transcription models, which are essentially individual translation models trained to translate from the transcribed text to the original writing system. This step can be trained on large quantities of monolingual data rather than being limited to parallel data, which is important if error propagation is to be kept to a minimum.

4 Data

We developed systems for four of the several language combinations taken into account for the general translation task. They are {cs,ru,uk}↔en and cs↔uk. We took part in the challenge under its constrained track, using only a portion of the data made available for the task. The following sections describe the data we used and how we processed and filtered it. We present the data sizes and their corresponding sources in Table 8 in Appendix A.

[2] Here and elsewhere in this paper, we use the so-called "scientific transliteration" when we transliterate Russian and Ukrainian. It is a standard, slightly language-dependent, transliteration scheme.

[3] Transliteration is generally defined as a bijective script change, a constraint that is too strict to allow for a significant increase in lexical overlap. Relaxing the bijectivity constraint, on the other hand, means that some information is lost. It is no longer a transliteration *stricto sensu*. Contrarily to (Muller et al., 2021), we therefore use the term *transcription* rather than transliteration to denote a transformation process that performs non-bijective changes.

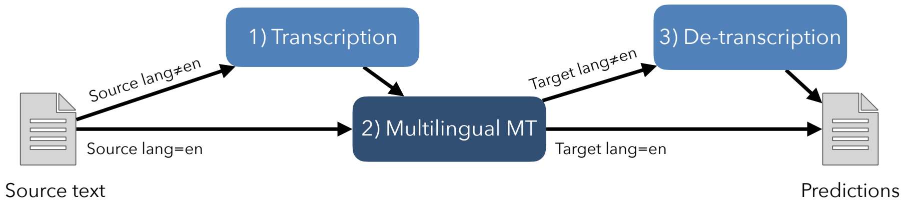


Figure 1: Illustration of our multilingual MT approach using common Slavic transcription.

4.1 Parallel Data

We used all of the parallel data provided for the language pairs we selected, with the exception of the back-translated news data, CzEng2.0 and two more datasets released at a later stage of the challenge, ELRC-EU acts, and Yakut parallel data, for the training of our NMT systems. We excluded the back-translated news data[4] and CzEng 2.0,[5] which are both back-translated data sources, after inspecting their respective content and discovering a large proportion of poorly translated sentences. To assess their quality and gauge the amount of noise present, the other parallel data were carefully examined. This was important especially for the web-mined data such as CCAligned, Wikimatrix, and CommonCrawl, which all contained a variety of quality issues identified in (Kreutzer et al., 2022).

Parallel Data Filtering: Each parallel corpus was subjected to a generic filtering pipeline involving the removal of blank lines and sentences without corresponding translations. We carried out language identification on the web-mined parallel corpora using FastText (Joulin et al., 2016a, 2017), thus removing sentence pairs where either the source or target is not in the intended language. Finally, the parallel corpora for each language pair were combined, and duplicate translation pairs were removed. Table 2 shows the original number of parallel sentences for the different language pairs and their corresponding sizes after filtering.

Language pair	Original	Filtered
cs–en	56,289,558	54,495,258
cs–uk	3,163,969	2,490,622
en–ru	31,052,852	25,584,007
en–uk	23,355,100	22,322,394

Table 2: Number of parallel sentences.

4.2 Monolingual Data

We used monolingual data to train the detranscription models. As with the parallel data, we removed empty lines, duplicated lines and also sentences that were not from the target language by doing language identification with FastText (Joulin et al., 2017, 2016b). This process was necessary since most of these sentences were web-mined text. The statistics of the monolingual data for each language are shown in Table 9 in Appendix A, along with their sizes before and after pre-processing.

For the transcription experiments, we randomly selected 20M sentences from the pre-processed monolingual texts for each of the Slavic languages.

4.3 Validation and Test Data

For each language pair, we chose 2000 and 3000 sentence pairs from the pre-processed parallel texts as our internal validation and test sets respectively, and the remaining sentences were used for training. In order to compare the various systems we developed, we also used the development set provided for the shared task (the FLORES development set and the WMT2018 test set depending on the language pair). This was also done for the systems with transcription. En↔uk and cs↔en models were only evaluated on the in-house test and the FLORES development sets because they were not in covered by the WMT2018 test sets. We also provide automatic scores on the WMT2022 test sets.

4.4 Subword Tokenisation

We tokenised all data using a joint SentencePiece (Kudo and Richardson, 2018) unigram model with a character coverage of 1.0 and a maximum sentence length of $4,096$ tokens. Specifically, for the bilingual systems, we uniformly sampled $5M$ monolingual sentences from the parallel training data of each language pair to have $10M$ sentences in total over which we trained a SentencePiece tokeniser. Similarly, for multilingual systems, we

[4] http://data.statmt.org/wmt20/translation-task/back-translation/

[5] https://ufal.mff.cuni.cz/czeng/czeng20

sampled a total of $10M$ monolingual sentences evenly from all monolingual data available for each language that the tokeniser was trained on.

5 Experiments and training

We submitted three categories of NMT systems: (i) the baseline bilingual translation models for each of the four language pairs in their original scripts, (ii) a multilingual model with common-Slavic transcription for {cs,uk,ru}→en, and (iii) a bilingual model with common-Slavic transcription for cs↔uk. Below, we provide details of these submitted systems, as well as the additional systems developed before and after the task's deadline.

5.1 NMT architecture and training

All models used the transformer-base architecture (Vaswani et al., 2017) within the Fairseq[6] toolkit (Ott et al., 2019). We use the `multilingual_translation` architecture for all models, except for those trained on a single language pair. We used batch sizes of $10,240$ tokens, a maximum sentence length of $1,024$, and a dropout of 0.3. For optimisation, we used Adam (Kingma and Ba, 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.998$, a learning rate of $1 * 5e^-5$ and a warm-up of $4,000$ updates. The optimiser uses a label-smoothed cross-entropy loss function with a label-smoothing value of 0.1. For multilingual models we use temperature sampling with $T = 1.5$. All models were trained until convergence based on the BLEU score on the development set. We use BLEU (Papineni et al., 2002) to evaluate our models and to choose the best checkpoints, calculated using SacreBLEU[7] (Post, 2018).

5.2 Baseline models

We trained a bilingual translation model for each of the four language pairs we covered. We chose a vocabulary size of $64k$ for all systems after experimenting with different sizes ($16k$, $32k$, and $64k$). We then fine-tuned each bilingual model to each of the two directions in the language pair (taking the best checkpoint of the bilingual model), resulting in a baseline model for each of the 8 translation directions.

We also trained a multilingual system for all of the language pairs, i.e. a single model that can translate in every direction, which was then finetuned to each language direction. We chose to use a vocabulary size of $64k$ based on the trends we found from the bidirectional model experiments. We chose not to go bigger in order to keep the model compact and comparable to the bilingual baselines, at least in terms of the number of parameters. Our comparison therefore tests whether for a same number of parameters multilingual models (and transcription) can be beneficial, despite the fact that multilingual vocabs are likely to result in a higher degree of segmentation for the individual languages.

5.3 Common Slavic transcription

To assess the impact of transcription, we trained bilingual and multilingual models on the transcribed versions of the Slavic parallel data. We follow the same setup as for the baseline models (i.e. bilingual/multilingual training and then fine-tuning on the specific language direction), simply substituting the original Slavic text with the transcribed versions.[8] When presenting the results, we refer to the transcribed version of Russian (ru), Czech (cs) and Ukranian (uk) as rl, cl and ul respectively.

Due to human error, our submitted multilingual systems were trained with a vocabulary of 16k rather than 64k, which severely penalised them and resulted in very low official scores. We report results with the intended vocabulary size of 64k in this article.

5.4 Detranscription models

For each Slavic language, we trained a detranscription model on $20M$ parallel sentences (transcribed→original), consisting of monolingual sentences and their automatically transcribed versions. We used a joint SentencePiece model of size $16k$ and used the same architecture as before. These models were applied after translation to make sure that transcribed Slavic outputs were in their original writing system.

6 Results

6.1 Baseline models (without transcription)

We first report results for our baseline models in Table 3 (i.e. without transcription).

We provide results for our in-house test set (from the same distribution as the training data), the FLORES devtest subset and the WMT2018 test

[6] `https://github.com/facebookresearch/fairseq`

[7] With the following parameters: `case:mixed|eff:no|tok:13a|smooth:exp|version:2.3.1`

[8] SentencePiece models were also retrained on the new data, keeping a vocabulary size of 64k.

	en→cs	**cs→en**	**cs→uk**	**uk→cs**	**en→ru**	**ru→en**	**en→uk**	**uk→en**
				Bilingual				
*In-house Test	43.11	45.38	39.16	40.20	42.66	47.07	33.32	38.16
FLORES$_{devtest}$	29.05	33.70	19.76	20.86	24.60	28.65	24.11	30.03
WMT 2018	20.81	29.03	–	–	23.77	28.15	–	–
WMT 2022	**33.62**	**39.45**	**27.40**	**25.65**	**23.60**	**34.71**	**20.72**	**34.42**
				Multilingual				
*In-house Test	36.02	39.42	27.08	28.50	35.71	40.74	34.19	38.97
FLORES$_{devtest}$	21.53	27.22	11.56	13.41	15.22	21.43	17.48	24.76
WMT 2018	15.36	21.18	–	–	15.04	21.19	–	–
WMT 2022	24.95	26.89	18.61	17.43	16.70	26.66	16.66	27.49

Table 3: BLEU score results for bilingual and multilingual baseline models (i.e. without transcription).

	en→cs	**cs→en**	**cs→uk**	**uk→cs**	**en→ru**	**ru→en**	**en→uk**	**uk→en**
				Bilingual				
In-house Test	41.72	44.94	35.80	38.04	38.67	46.50	28.89	37.69
FLORES$_{devtest}$	28.83	33.56	19.05	20.09	21.77	27.93	21.94	28.86
WMT 2018	20.66	27.64	–	–	21.64	27.57	–	–
WMT 2022	33.42	37.83	26.43	24.96	21.22	34.43	18.69	32.7
				Multilingual				
In-house test	36.75	40.08	30.64	32.99	34.57	41.19	29.73	38.71
FLORES$_{devtest}$	22.34	28.15	15.90	16.22	17.96	22.64	20.62	24.78
WMT 2018	15.85	22.39	–	–	17.85	22.24	–	–
WMT 2022	26.08	29.00	23.20	21.15	17.66	27.32	18.36	28.48

Table 4: BLEU score results for bilingual and multilingual models using transcription for all Slavic languages.

set (when available). Although the BLEU scores are not directly comparable across test sets, the baseline results are generally quite high. The highest results are seen for cs↔en and for all sets other than the WMT2022 test set, the lowest are generally seen for cs↔ul, which correspond to the highest and lowest resourced language pairs respectively. Interestingly, the en↔uk test set are comparatively tougher than the other sets we evaluate with.

When we compare bilingual and multilingual results, it is clear that the bilingual models are largely superior for all language directions, with very large differences in BLEU scores across evaluate sets. The only BLEU scores that are higher for the multilingual model is for en↔uk, for which the in-house test set gives slightly higher results. However, this does not hold for the other test sets, indicating overfitting of the models. These results are not so surprising given the relatively small vocabulary size of 64k for the four languages included in training. This is to compare with the bilingual models' 64k vocabulary sizes spread over two languages only. The obligation to share a same vocabulary size amongst more languages (and more scripts) is certainly not compensated by any gain that could possibly be had through multilingual transfer.

6.2 Results with transcription

In Table 4 we provide the results of bilingual and multilingual models with transcription (and detranscription where necessary).

Although transcription should not help the bilingual models that translate to and from English since there is only one Slavic language involved, we include these results for comparative purposes. Ideally, these results (for {cs,uk,ru}↔en) should be identical to the baseline results, showing that transcription does not introduce noise into the process. In reality, we see a systematic drop in results when transcribing for into-English directions, and a greater drop in BLEU score for into-Slavic directions, most likely due to errors introduced by the detranscription model. Interestingly, some directions suffer much more than others (e.g. en→uk and en→ru have a drop of over 2 BLEU vs. en→cs's drop of 0.20 BLEU on WMT2022). This could well be a reflection of the fact that the transcription scheme was centred around Czech, with fewer modifications being made to this language than to the others.

For the multilingual models, the scores are again much lower than the bilingual models with transliteration for all directions, although some slight improvements are seen for into-English directions, although the performance is much closer for en→uk. We do however see an improvements across the board on the results of the baseline multilingual models (i.e. without transcription), suggesting that transcribing helps to marginally make up some of the lost scores. Unfortunately, it is unclear whether this is due to the vocabulary now being spread over fewer different scripts or whether transcription does help provide better transfer in some other ways.

7 Discussion

Given these disappointing results, it is important to make a first step to understanding why transcription does not help. We therefore look at some additional results concerning the noise that the transcription step might be introducing: (i) the translation results for the detranscription step itself and (ii) comparative results for cs↔uk when transcribing the source, the target or both.

Detranscription quality We show the results for the detranscription step itself in Table 5, where we apply our detranscription models to the texts to which our transcription rules have been applied. The BLEU scores are very high, but not exactly perfect, suggesting that errors are being introduced in this step. The results are highest for Czech, therefore confirming our earlier hypothesis that this step is degrading less for this language given that fewer changes are made.

We also provide results (Table 6) of the raw output of the from-English bilingual models with transcription (i.e. before applying detranscription). We compare these to the results of the bilingual baselines (trained to produce the correct script) but with automatic transcription applied to the outputs in order to provide a point of comparison in terms of the BLEU score. The results are lower for the bilingual models with transcription for Russian and Ukrainian, suggesting that the outputs of the MT models are also far from perfect, and that transcription may be introducing ambiguities and making it harder for the models to learn. However, as can be seen in previous results, the same cannot be said for Czech, where the results are actually slightly higher for the bilingual model with transcription compared to the bilingual baseline with transcription applied.

	cl→cs	**rl→ru**	**ul→uk**
FLORES$_{devtest}$	97.49	94.74	96.29
WMT 2022 (src)	96.47	95.34	94.70
WMT 2022 (ref)	97.33	96.24	97.12

Table 5: BLEU score results for detranscription.

	en→cl	**en→rl**	**en→ul**
Bilingual with transcription			
FLORES$_{devtest}$	29.53	22.90	22.62
WMT 2022	34.06	22.56	19.27
Transcribing bilingual baseline's output			
FLORES$_{devtest}$	29.37	25.35	24.60
WMT 2022	33.87	23.75	20.94

Table 6: Comparison of bilingual models with transcription and of baseline bilingual models with transcription applied to the outputs. Results (BLEU scores) are provided on transliterated references.

Comparative results for cs↔uk with different combinations of transcription Table 7 shows the results for cs↔uk when transcribing the source, target or both. The best results when using the original scripts, as can be seen in previous results. However, the results suggest that in some scenarios, it could be better to transcribe just the source rather than to transcribe both source and target. The advantage of this for uk→cs is that Ukrainian is being made to look more like Czech, but without there being extra errors added by the detranscription step. (Muller et al., 2021) showed that transcribing could be useful for lower-resource languages, so a possibility here is that the languages are sufficiently high-resource for transcription not to help so much and for the errors introduced in detranscription to outweigh any potential benefits.

	none	source	target	both
cs→uk				
FLORES$_{devtest}$	19.76	19.64	19.32	19.05
WMT 2022	27.40	27.01	26.82	26.43
uk→cs				
FLORES$_{devtest}$	20.86	20.41	18.50	20.09
WMT 2022	25.65	25.15	25.41	24.96

Table 7: BLEU score results for bilingual cs↔uk models when transliterating neither source nor target (none), just the source, just the target or both. Results are shown after detranscription.

8 Conclusion

Setting aside the fact that multilingual models provide very inferior results to specific bilingual models for the same number of parameters, our results suggest that the answer to the question “Does transcription help cross-script machine translation?” is no. This is at least for the languages on which we experimented and given the amount of training data we had at our disposition. Our bilingual model results show that transcription harms performance, whether it is done on the source side, the target side or both sides. There are several possible explanations for this: (i) the relatively high-resource scenario we are working in, where baselines can already achieve good results and where little gain can be achieved through this type of transfer, (ii) the possibility that transcription introduced ambiguities that could harm translation, and (iii) the detranscription step itself also introducing errors.

Acknowledgements

This work was partly funded by Rachel Bawden’s and Benoît Sagot’s chairs in the PRAIRIE institute funded by the French national agency ANR as part of the “Investissements d’avenir” programme under the reference ANR-19-P3IA-0001 and also by the Emergence project, DadaNMT, funded by Sorbonne Université.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Rachel Bawden, Nikolay Bogoychev, Ulrich Germann, Roman Grundkiewicz, Faheem Kirefu, Antonio Valerio Miceli Barone, and Alexandra Birch. 2019. The University of Edinburgh’s submissions to the WMT19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 103–115, Florence, Italy. Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Bernardo Stearns, Arun Jayapal, Sridevy S, Mihael Arcan, Manel Zarrouk, and John P McCrae. 2019. Multilingual multimodal machine translation for Dravidian languages utilizing phonetic transcription. In *Proceedings of the 2nd Workshop on Technologies for MT of Low Resource Languages*, pages 56–63, Dublin, Ireland. European Association for Machine Translation.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Vikrant Goyal, Sourav Kumar, and Dipti Misra Sharma. 2020. Efficient neural machine translation for low-resource languages via exploiting related languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 162–168, Online. Association for Computational Linguistics.

Thanh-Le Ha, Jan Niehues, and Alex Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. In *Proceedings of the 13th International Conference on Spoken Language Translation*, Seattle, Washington D.C. International Workshop on Spoken Language Translation.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou, and Tomás Mikolov. 2016b. Fasttext.zip: Compressing text classification models. *CoRR*, abs/1612.03651.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252, Brussels, Belgium. Association for Computational Linguistics.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol

Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroro Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2022. Quality at a Glance: An Audit of Web-Crawled Multilingual Datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual NMT representations at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1565–1575, Hong Kong, China. Association for Computational Linguistics.

Anoop Kunchukuttan and Pushpak Bhattacharyya. 2021. *Machine Translation and Transliteration Involving Related and Low-resource Languages*. Chapman and Hall/CRC.

Hairong Liu, Mingbo Ma, Liang Huang, Hao Xiong, and Zhongjun He. 2019. Robust neural machine translation with joint textual and phonetic embedding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3044–3049, Florence, Italy. Association for Computational Linguistics.

Mieradilijiang Maimaiti, Yang Liu, Huanbo Luan, and Maosong Sun. 2019. Multi-Round transfer learning for Low-Resource NMT using multiple High-Resource languages. *ACM Transactions on Asian Low-Resource Language Information Processing*, 18(4):1–26.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

Preslav Nakov and Jörg Tiedemann. 2012. Combining word-level and character-level models for machine translation between closely-related languages. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 301–305, Jeju Island, Korea. Association for Computational Linguistics.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

José Carlos Rosales Núñez, Djamé Seddah, and Guillaume Wisniewski. 2019. Phonetic normalization for machine translation of user generated content. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 407–416, Hong Kong, China. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Simeng Sun, Angela Fan, James Cross, Vishrav Chaudhary, Chau Tran, Philipp Koehn, and Francisco

Guzmán. 2022. Alternative input signals ease transfer in multilingual machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5291–5305, Dublin, Ireland. Association for Computational Linguistics.

Svetlana Tchistiakova, Jesujoba Alabi, Koel Dutta Chowdhury, Sourav Dutta, and Dana Ruiter. 2021. EdinSaar@WMT21: North-germanic low-resource multilingual NMT. In *Proceedings of the Sixth Conference on Machine Translation*, pages 368–375, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

A Data sources

Tables 8 and 9 give the amount of data for each data source in the parallel and monolingual data respectively.

Source	**en-ru**	**en-cs**	**en-uk**	**cs-uk**
AirBaltic	1092			
ECB		3100		
CZECHTOURISM	7328			
RAPID		263287		
EMA		495234		
EESC		1329010		
UNCorpus4	23239280			
NEWS Commentary	333899	253639		
WorldBank	25849		1628	
Paracrawl	5377911	50632492	13354365	
WikiTitles	1189107	410978		
WikiMatrix		2094650		
EUROPARL		645330		
Commoncrawl	878386	161838		
Opus				
Bible			15901	7953
Open Subtitles			877780	730804
EUBooks			1793	1506
TED2020			208141	115351
Wikimedia			348143	1959
MultiCCAligned			8547349	2306396
Dev set				
FLORES (dev)	997		997	
FLORES (devtest)	1012		1012	
NEWSTEST2018	991			

Table 8: Parallel data sources.

Source	**Cs**	**En**	**Ru**	**Uk**
News crawl	12203274	39361312	15441304	411439
Europarl v10	669676			
News Commentary	282139	660667	404978	
Common Crawl	333498145	-	1168529851	
UberText Corpus				-
fiction				1811548
news				31021650
ubercorpus				48620146
wikidump				15786948
Leipzig Corpora	-	-	-	-
ukr_mixed_2012				1000000
ukr_news_2020				1000000
ukr_newscrawl_2018				1000000
ukrua_web_2019				1000000
ukr_wikipedia_2021				1000000
Legal Ukrainian				7568246
Common Crawl (filt.)	275825036	-	1150607428	-
Total (concat.)	293980125	-	1170453710	110219977
Total (dedup.)	290477308	-	1155825622	53177077

Table 9: Monolingual data sources.

NAIST-NICT-TIT WMT22 General MT Task Submission

Hiroyuki Deguchi[†,††] **Kenji Imamura**[††] **Masahiro Kaneko**[†††] **Yuto Nishida**[†]
Yusuke Sakai[†] **Justin Vasselli**[†] **Huy Hien Vu**[†] **Taro Watanabe**[†]
[†] Nara Institute of Science and Technology [†††] Tokyo Institute of Technology
[††] National Institute of Information and Communications Technology
[†]{deguchi.hiroyuki.db0, nishida.yuto.nu8, sakai.yusuke.sr9,
vasselli.justin_ray.vk4, vu.huy_hien.va9, taro}@is.naist.jp
[††]kenji.imamura@nict.go.jp, [†††]masahiro.kaneko@nlp.c.titech.ac.jp

Abstract

In this paper, we describe our NAIST-NICT-TIT submission to the WMT22 general machine translation task. We participated in this task for the English ↔ Japanese language pair. Our system is characterized as an ensemble of Transformer big models, k-nearest-neighbor machine translation (kNN-MT) (Khandelwal et al., 2021), and reranking.

In our translation system, we construct the datastore for kNN-MT from back-translated monolingual data and integrate kNN-MT into the ensemble model. We designed a reranking system to select a translation from the n-best translation candidates generated by the translation system. We also use a context-aware model to improve the document-level consistency of the translation.

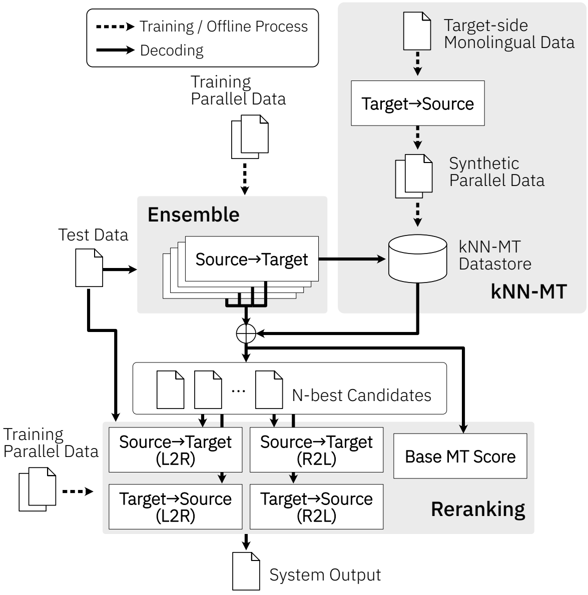


Figure 1: System overview.

1 Introduction

We participated in the WMT22 general machine translation task in two language directions, English-to-Japanese (En-Ja) and Japanese-to-English (Ja-En). We built our system on an ensemble of Transformer big models, k-nearest-neighbor machine translation (kNN-MT) (Khandelwal et al., 2021), and reranking. Figure 1 shows an overview of our system.

Our translation system is a combination of kNN-MT and an ensemble of four Transformer big models. We train each of the Transformer NMT models using a different random seed, and pick one model as kNN-MT. A notable point about our system is that we construct the datastore for kNN-MT from back-translated monolingual data rather than reusing training data. We found that using back-translated data improves translation performance compared with using a parallel training corpus for the datastore.

Our reranker is designed to select the translation from the n-best translation candidates generated by the translation system. The reranker computes the weighted sum of each translation candidate across multiple models and selects the translation candidate with the highest score. We used k-best batch MIRA (Cherry and Foster, 2012) to select the weights for the model scores that maximize the BLEU score of the development set.

2 Corpora and Preprocessing

For the training data, we used all the provided bilingual parallel data: JParaCrawl v3 (Morishita et al., 2020), News Commentary v16 (Tiedemann, 2012), Wiki Titles v3, WikiMatrix, Japanese-English Subtitle Corpus (Pryzant et al., 2018), The Kyoto Free Translation Task Corpus, and TED Talks. As the English translation of the Japanese-English Subtitle Corpus is only available in lowercase, we trained a Moses truecaser (Koehn et al., 2007) using the other corpora to add capitalization into the subtitle corpus. After truecasing, the first letter of each sentence was capitalized using detruecasing to produce sentence-cased English text

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 244 - 250
December 7-8, 2022 ©2022 Association for Computational Linguistics

to match the casing in the other corpora. We cleaned the data by removing duplicate lines and applying language filtering. As much of the training data was crawled from the internet, we used fasttext (Joulin et al., 2016a,b) to predict the language of each sentence and removed the sentences that were not predicted to be the correct language. This has the effect of reducing the noise of the dataset by removing sentences with garbage tokens. We tokenized the text into subword units using a joint vocabulary size of 64,000, a character coverage of 99.98%, and byte fallback using sentencepiece (Kudo and Richardson, 2018). After subword segmentation, all sentences shorter than 1 token or longer than 250 tokens were removed. We also removed all sentences where the number of tokens in one language was more than double the number of tokens in the translation, i.e the ratio of tokens between the source and target is >2.0. After filtering, 27,784,519 sentence pairs remained for training.

3 Translation System

3.1 Base Model

Our translation model is based on the Transformer big architecture (Vaswani et al., 2017) with an FFN size of 8,192 implemented in FAIRSEQ (Ott et al., 2019). The hyperparameters for our translation models are shown in Table 1.

Translation Model	
Architecture	Transformer big with FFN size of 8,192
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$)
Learning Rate Schedule	Inverse square root decay
Warmup Steps	4,000
Max Learning Rate	0.001
Dropout	0.3
Gradient Clipping	1.0
Label Smoothing	$\epsilon_{ls} = 0.1$
Mini-batch Size	512,000 tokens
Number of Updates	80,000 steps
Averaging	Save checkpoint for every 1,000 steps and take an average of last 10 checkpoints
Length Penalty	1.0
Beam Size	10
Reranker Model	
Architecture	Transformer big with FFN size of 8,192
Optimizer	Adam ($\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 1 \times 10^{-8}$)
Learning Rate Schedule	Inverse square root decay
Warmup Steps	4,000
Max Learning Rate	0.001
Dropout	0.3
Position Embeddings	SHAPE ($K = 512$)
Gradient Clipping	1.0
Label Smoothing	$\epsilon_{ls} = 0.1$
Mini-batch Size	512,000 tokens
Number of Updates	80,000 steps
Averaging	Save checkpoint for every 2,000 steps and take an average of last 10 checkpoints

Table 1: Hyperparameters of our translation and reranker models.

3.2 kNN-MT

kNN-MT (Khandelwal et al., 2021) extends the decoder of a trained machine translation model using the k-nearest-neighbor search algorithm, and retrieves the cached translation examples. The method consists of two steps, *datastore creation*, which creates key-value translation memory, and *generation*, which calculates an output probability distribution based on the nearest neighbors of cached translation memory.

Datastore creation The typical NMT model is composed of an encoder that encodes the source sentence X and a decoder that generates target tokens $Y = (y_1, y_2, \dots, y_I)$. Each target token y_i is generated based on its output probability $P(y_i|X, y_{<i})$. kNN-MT caches parallel text $\mathcal{D}$ in a datastore represented as key-value memory $\mathcal{M}$. The value is a token y_i that comes from a target sentence in a parallel corpus, and the key is its intermediate vector h_i of each time step computed by the decoder. The datastore is formulated as follows:

$$\mathcal{M} = \{(h_i, y_i), \forall y_i \in Y \mid (X, Y) \in \mathcal{D}\}. \quad (1)$$

In our model, we use the 1024-dimensional vector representation from the decoder before it is passed to the final feed-forward network as the key h_i.

We use FAISS (Johnson et al., 2019), which is a toolkit for kNN search, to represent the datastore and search for the nearest neighbors. We use the `OPQMatrix` vector transform, `IndexIVFPQ` index, and `IndexFlatL2` index as the coarse quantizer. The hyperparameters of our search index are shown in Table 2.

Generation During decoding, kNN-MT generates output probabilities by computing the linear interpolation between the kNN and MT probabil-

Type	Value
Shape of OPQ matrix	$\mathbb{R}^{1024\times1024}$
Number of clusters (IVF)	65,536
Number of sub-vectors (PQ)	64
Number of clusters to search	64
Number of top-k neighbors	16

Table 2: Hyperparameters of our search index.

ity distributions,

$$P(y_i|X, y_{<i}, \theta) = \lambda p_{\text{kNN}}(y_i|X, y_{<i}, \theta) + (1-\lambda)p_{\text{MT}}(y_i|X, y_{<i}, \theta), \tag{2}$$

where λ is a hyperparameter for weighting each probability and θ represents the trained weight parameters and we set $\lambda = 0.4$

The k-nearest-neighbor keys $\mathcal{N}$ are converted into a distribution over the vocabulary p_{kNN} by applying softmax function.

$$p_{\text{kNN}}(y_i|X, y_{<i}, \theta) \propto \sum_{(k_j, v_j)\in\mathcal{N}} \mathbb{1}_{y_i=v_j} \exp\left(\frac{-||k_j - h_i||_2^2}{\tau}\right), \tag{3}$$

where k_j and v_j are the top-j neighbor key and value respectively and τ is a hyperparameter that represents the temperature of softmax and we set $\tau = 100$.

3.3 Back-Translated Monolingual Datastore

Our kNN-MT system uses back-translated monolingual data for the datastore instead of bilingual corpora. This method allows us to use monolingual resources without any additional training. First, we use the bilingual corpora to train a target-to-source translation model, which is then used to back-translate the monolingual corpora. The back-translated synthetic source sentences are then passed through the source-to-target model, which generates the intermediate vectors for each decoding time step to fill the datastore.

The monolingual data for English was taken from News Commentary v16 (Tiedemann, 2012), Europarl v10, Leipzig's news corpora (2018-2020), news-typical (2016), newscrawl and newscrawl-public (2018), web and web-public (2018-2020), and the largest available size of wikipedia corpus from each year for a total of over 26 million sentences. The monolingual data for Japanese includes News Commentary v16 (Tiedemann, 2012), and all Leipzig news, newscrawl, web, web-public, and wikipedia corpora from 2014 to 2021, totaling over 15 million sentences.

	Japanese	English
# of sentences	15,051,874	26,237,110
# of tokens	396,647,042	690,734,548

Table 3: Monolingual data statistics.

We preprocessed the monolingual data much the same way as the bilingual data, removing duplicate lines and using fasttext to filter out sentences where the predicted language did not match the target language. The text was tokenized into subword units using the model trained on the bilingual corpora. In order to reduce the time for the back-translation, sentences with more than 200 tokens were removed from the monolingual data. Table 3 shows the monolingual data statistics after preprocessing. Note that the number of tokens is equal to the size of the resulting kNN datastore for each target language.

3.4 kNN-MT with Ensemble Model

We integrate kNN-MT into the ensemble model. We train Transformer big models with different random seeds, just as we would build a normal ensemble model. Because the kNN search is too computationally expensive, we randomly pick a model instance and use it for the search as follows:

$$\begin{aligned} &P(y_i|X, y_{<i}, \theta_1, \dots, \theta_M) \\ &= \lambda p_{\text{kNN}}(y_i|X, y_{<i}, \theta_1) \\ &+ \frac{1-\lambda}{M}\sum_{m=1}^{M} p_{\text{MT}}(y_i|X, y_{<i}, \theta_m), \end{aligned} \tag{4}$$

where M is the number of model instances for the ensemble, and we set $M = 4$.

4 Reranker

Our reranker selects one of the n-best translation candidates from the translation system. Similar to other rerankers, it computes the weighted sum of multiple model scores for each translation candidate and selects the candidate with the maximum score. We used the average log-likelihoods of the

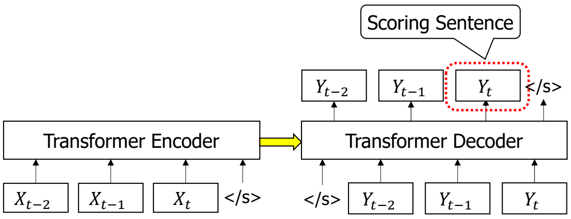


Figure 2: Context-aware model (context length $\ell = 2$).

words in each document as the model scores:

$$\hat{D} = \operatorname*{argmax}_{Y_{t=1}^{T}} \left\{ \sum_k w_k \frac{\sum_{t=1}^{T} \sum_{i=1}^{|Y_t|} \mathcal{L}_k(y_{t,i})}{\sum_t |Y_t|} \right\}, \tag{5}$$

where D denotes a set of document translations, which consists of T translations ($D = Y_{t=1}^{T}$), and Y_t is the t-th translation in the document. $y_{t,i}$ denotes the i-th token in the translation Y_t, in which the number of tokens is $|Y_t|$. $\mathcal{L}_k(y_{t,i})$ denotes the log-likelihood of the token $y_{t,i}$ scored by the k-th model, and w_k is the weight of the k-th model.

The weights of the model scores were trained to maximize the BLEU score of the development set. We used k-best batch MIRA (Cherry and Foster, 2012) to optimize the weights.

4.1 Reranking Models

A characteristic of our reranker is the use of context-aware model scores, which are the log-likelihoods calculated per document of the test (or development) set. By taking the context into consideration during scoring, we expected to improve the consistency of the translation throughout each document.

We use an N-to-N translation model of Tiedemann and Scherrer (2017) as the context-aware model, which translates multiple concatenated sentences. Figure 2 illustrates the context-aware model in which the context length is two sentences ($\ell = 2$). The model computes the log-likelihood of the target translation Y_t using preceding ℓ sentences; that is,

$$\mathcal{L}_k(y_{t,i}) = \log p_k(y_{t,i} | X_{t-\ell}^{t}, Y_{t-\ell}^{t}), \tag{6}$$

where $p_k(\cdot)$ denotes the likelihood computed by the k-th model, and $X_{t-\ell}^{t}$ and $Y_{t-\ell}^{t}$ denote the source sentences and their translations from $t - \ell$ to t, respectively.

We used five models in total: the score from our translation system, and a combination of source-to-target and target-to-source translation and left-to-right (L2R) and right-to-left (R2L) decoding directions.

We trained the R2L models by reversing the order of the target tokens. Although the order is the same during scoring, we reversed the target tokens after concatenating multiple sentences. Therefore, the sentence order of the target side becomes (Y_t, Y_{t-1}, Y_{t-2}). Note that the scoring sentence is the last sentence (Y_{t-2}), and the R2L models score sentences later than the L2R models.

4.2 Training and Reranking

We used only the parallel corpora described in Section 2 and trained Transformer big models (Vaswani et al., 2017) with an FFN size of 8,192 for reranking. However, the trained models were sentence-wise models because we did not use document information in the training corpora. To apply the sentence-wise models to the N-to-N translation, we modified it using the following techniques.

- We did not use sentence separators (e.g., '[SEP]' between sentences) because the sentence-wise models did not include such separators.

  In the reranking task, we know the target tokens in advance, and we can easily identify the tokens for the target sentence, which we are scoring, without the separators.

- In the N-to-N translation, we had to score long translations because we simply concatenated multiple sentences during inference using a sentence-wise model which was not aware of the concatenated sentences. To learn appropriate models for long translations from sentences, we used the shifted absolute position embeddings (SHAPE) (Kiyono et al., 2021) to make a model invariant to absolute positions. The maximum shift was 512.

For the hyperparameters for the reranking models, we used the same setting as Kiyono et al. (2020) (Table 1).

To search for the best translations while considering context, we applied the beam search method to search for the translations that satisfied Eq. (5). We used a beam width of 10.

	En-Ja	Ja-En
Single Model	23.17	24.67
+ Ensemble	23.82	25.41
+ kNN-MT	24.43	25.16
+ kNN-MT with Ensemble	**24.72**	**25.84**

Table 4: Ablation study of our translation system on newstest 2020 (BLEU %).

Datastore	En-Ja	Ja-En
No Datastore (w/o kNN-MT)	23.17	24.67
Training Data	22.76	24.79
BT Monolingual Data	**24.43**	**25.16**

Table 5: Comparison of the kNN-MT datastore on newstest 2020 (BLEU %)

		newstest	
Direction	Reranking	2020	2021
En-Ja	No Reranking	24.7	26.8
	$\ell = 0$	24.8	27.3
	$\ell = 2$ (submission)	24.9	**27.4**
	$\ell = 4$	**25.0**	27.3
	Oracle	29.6	31.8
Ja-En	No Reranking	**25.6**	22.5
	$\ell = 0$	25.5	22.7
	$\ell = 2$ (submission)	25.5	**22.8**
	$\ell = 4$	25.5	22.7
	Oracle	29.8	25.7

Table 6: BLEU scores according to the reranking method.

5 Results

5.1 Translation System

Ablation Study To validate the effectiveness of our translation system, we performed an ablation experiment. Table 4 shows the experimental result on newstest 2020. Note that this experiment does not use a reranker system, and we evaluated the 1-best translation. The result shows that both ensemble and kNN-MT are effective, and combining them further improves translation performance.

kNN-MT Datastore As noted in Section 3.3, our kNN-MT datastore uses different data than the training corpus. We evaluated the translation performance of kNN-MT on a single Transformer model without ensemble. Table 5 shows the comparison of the kNN-MT datastore evaluated on newstest 2020. In the table, 'No Datastore' indicates that kNN-MT is not used, and 'Training Data' and 'BT Monolingaul Data' indicate that the datastore is constructed from training data and back-translated monolingual data, respectively. As shown in the table, our 'BT Monolingual Data' datastore outperforms the datastore constructed from the training data, despite its smaller size.

5.2 Context Length at Reranking

Table 6 shows the BLEU scores according to the reranking method. 'No Reranking' indicates the best translations output from the translation system. 'Oracle' always chooses the translation with the highest sentence BLEU score from the n-best translation candidates and represents the output of a perfect reranking system. The other cases indicate the BLEU scores of our reranker of varying context lengths ℓ.

The results show that our reranker improved the BLEU scores from the 'No Reranking' case, except for the case of Ja-En in newstest2020. However, the context length did not affect the BLEU scores. (We submitted the case of $\ell = 2$.) The BLEU scores of the reranker were still lower than that of 'Oracle', and future work will include studying the context-aware models to improve it further.

5.3 Placeholders

This year, the test set for the General MT task contained a set of placeholder tags, which should be output without translation. However, the provided parallel corpora for the task did not contain these special tokens. To solve this problem, we built a training set with placeholders using the existing parallel corpora.

We focused on the WikiTitles corpus, which is a subset of the parallel corpora provided for the General MT task. Most bitexts in WikiTitles are named entities because the corpus was extracted from Wikipedia titles. We substituted the parts that matched the WikiTitles entries with the placeholders.

In detail, we only extracted WikiTitles entries of five characters or more in Japanese and 10 characters or more in English from the parallel cor-

			Translation	
Direction	Placeholder (PLH)	Source	Base	PLH
En-Ja	#NAME#	3	2	3
	#NUMBER#	2	1	2
	#PRS_ORG#	55	52	55
	#URL#	4	3	4
	BLEU	-	39.2	38.5
Ja-En	#Organization1#	1	0	1
	#Person#	1	1	1
	#Product1#	116	79	116
	#Product2#	42	26	40
	#Product3#	7	4	7
	#Product4#	2	2	2
	#Product5#	2	2	2
	#Product6#	2	2	2
	#URL#	5	5	5
	#URL1#	1	0	1
	BLEU	-	22.7	22.7

Table 7: Results of placeholder translation.

pora using the longest match, and substituted the matched parts with the placeholders. (We used only one placeholder type: '#PLACEHOLDER#'.) Note that we excluded parallel sentences in Wiki-Titles from the parallel corpora in advance. As a result, we obtained additional 1.7 million parallel sentences that contained the placeholders and used them for training and fine-tuning.

During fine-tuning the translation system, we set an unused token in the vocabulary to the placeholder tag, and fine-tuned our translation models with a combination of data from the original training data (without placeholders) and the new data with the placeholders for two additional epochs from the averaged checkpoints.

Our placeholder corpus contained only a single placeholder tag instead of the rich variety of tags contained in the test set. We resolved this during the translation of the test set by first replacing all placeholder tags with our placeholder (#PLACEHOLDER#) before translation. After translation, we identified and replaced our #PLACEHOLDER# tag with the original tag from the source sentence. In the case of multiple placeholder tags in the same sentence, we preserved the original order when converting them back into the test placeholder tag set.

Table 7 shows the results of placeholder translation; that is, the number of placeholders and the BLEU score for the wmttest2022 test set. 'Base' and 'PLH' indicate translation using the model without/with fine-tuning on the placeholder corpus, respectively. The 'Base' model failed to translate some placeholders because it processed the placeholders as strings and translated them after subword segmentation. By contrast, the 'PLH' model translated the placeholders almost perfectly. However, the model fine-tuned on the placeholder corpus did not improve the BLEU scores, and we submitted the result of the 'Base' model.

6 Conclusions

In this paper, we described our submission of the joint team of NAIST, NICT, and TIT (NAIST-NICT-TIT) to the WMT22 general MT task. We participated in this task for the En ↔ Ja translation. Our system is built on an ensemble of Transformer big models, kNN-MT with using monolingual data, and k-best batch MIRA reranker. We would like to investigate each method and further improve translation performance.

References

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *International Conference on Learning Representations (ICLR)*.

Shun Kiyono, Takumi Ito, Ryuto Konno, Makoto Morishita, and Jun Suzuki. 2020. Tohoku-AIP-NTT at WMT 2020 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 145–155, Online. Association for Computational Linguistics.

Shun Kiyono, Sosuke Kobayashi, Jun Suzuki, and Kentaro Inui. 2021. SHAPE: Shifted absolute position embedding for transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3309–3321, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, abs/1808.06226.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2020. JParaCrawl: A large scale web-based English-Japanese parallel corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3603–3609, Marseille, France. European Language Resources Association.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

R. Pryzant, Y. Chung, D. Jurafsky, and D. Britz. 2018. JESC: Japanese-English Subtitle Corpus. *Language Resources and Evaluation Conference (LREC)*.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Samsung R&D Institute participation in WMT 2022 General MT Task

Adam Dobrowolski[1], **Mateusz Klimaszewski**[*2], **Adam Myśliwy**[1], **Marcin Szymański**[1],
Jakub Kowalski[1], **Kornelia Szypuła**[1], **Paweł Przewłocki**[1], **Paweł Przybysz**[1]
[1]Samsung R&D Institute, Warsaw, Poland
[2]Warsaw University of Technology, Warsaw, Poland
{a.dobrowols2, a.mysliwy, m.szymanski, j.kowalski5, k.szypula, p.przybysz}@samsung.com
mateusz.klimaszewski.dokt@pw.edu.pl, p.przewlocki@partner.samsung.com

Abstract

This paper presents the system description of Samsung R&D Institute Poland participation in WMT 2022 for General MT solution for medium and low resource languages: Russian and Croatian. Our approach combines iterative noised/tagged back-translation and iterative distillation. We investigated different monolingual resources and compared their influence on final translations. We used available BERT-like models for text classification and for extracting domains of texts. Then we prepared an ensemble of NMT models adapted to multiple domains. Finally we attempted to predict ensemble weight vectors from the BERT-based domain classifications for individual sentences. Our final trained models reached quality comparable to best online translators using only limited constrained resources during training.

1 Introduction

Samsung R&D Institute Poland (SRPOL) participated in the WMT 2022 General MT task for three translation directions: EN→RU, RU→EN and EN→HR. All our systems were built using only constrained datasets. In contrast to previous years, where the task focused on news translation, this year's task was domain-independent. However, MT models benefit a lot from domain adaptation. Therefore, we decided to prepare an ensemble of NMT models adapted to multiple domains to benefit from domain adaptation and improve generalization. We prepared a news profiled model but also a general-purpose one. Additionally, we worked on medical and legal domains; however, there was very limited in-domain data in the constraint path for this domains and we had to extract pseudo in-domain data from monolingual corpora.

Our system was implemented using Marian framework. The core of the submitted solution is iterative back-translation and iterative distillation combined with finetuning and ensembling. Besides, we used BERT models for data filtering to prepare corpora for training domain-adapted models. Finally, we created dynamic ensemble weighting to choose the best combination of single models in the final translations. All techniques combined allowed us to improve baseline models by 3-6 BLEU (Papineni et al., 2002) and reach the quality comparable with online translators (measured by BLEU).

2 System overview

2.1 MT model

Our models were trained with the Marian NMT (Junczys-Dowmunt et al., 2018) toolkit. We used Marian for training, back-translation, noise generation, language models and data filtering.

The training was performed on a *transformer-big* model (embedding dimension of 1024 and a feed-forward layer dimension of 4096) (Vaswani et al., 2017). We experimented with different sizes of models and different configurations of encoder-decoder layers, but we achieved no significant improvement over the default *transformer-big* configuration. Most models had a setup of either 7-5 or 8-4 encoder-decoder layers. Best single models were trained with FF layer dimension 6144, but the improvement was marginal – 0.1 BLEU better than the default dimension of 4096.

Our training used batches of size 256GB (8xGPU, 32GB workspace). The optimizer was Adam (Kingma and Ba, 2015) with a learning rate of 0.0003 and linear warm-up for the initial 40 000 updates with subsequent inverted squared decay. A few initial EN↔RU training were regularized with dropout 0.1, but the following did not use any dropout. All training for EN→HR had the dropout set to 0.1.

2.2 Iterative training process

Iterative back-translation (Hoang et al., 2018) is a known technique of improving performance of

[*] Work done while at Samsung R&D Institute Poland.

MT models. Iterative distillation approach applied by NiuTrans (Zhou et al., 2021) allowed them to achieve impressive results in WMT21. During our work we combined both techniques in parallel during each iteration.

First baseline models were trained using only provided parallel corpora. Further training iterations were enriched with back-translation (iterative back-translation). With each iteration we used new back-translation prepared by best ensembles translating from target to source.

After a few iterations of iterative back-translation we started iterative distillation. Training corpus was enriched with corpora distilled from best ensembles. (→ 3.3). As a result the whole corpus consisted of parallel part, back-translated part and distilled part.

After training iteration converged we finalized the iteration with additional tuning using parallel corpora or specialized tuning corpora (→ 3.4). After the tuning we selected a new best ensemble containing the new trained model. The best ensemble was chosen by selecting the best performing on Flores devtest (Goyal et al., 2022) and Newstest 2021. With this new ensemble we prepared new back-translation and a new distilled corpus for next iterations.

Algorithm 1 Iterative training process

```
procedure ITERATEDTRAININGS
    $M_{enru}$ ←train($bitext_{enru}$)
    $M_{ruen}$ ←train($bitext_{ruen}$)
    while $models$ not converged do

        $bktr$ ←translate($mono_{en}, M_{enru}$)
        $dist$ ←distill($bitext_{ruen}, M_{ruen}$)
        $corpus = bktr + dist + bitext_{ruen}$
        $model_{ruen}$ ←train($corpus$)
        $model_{ruen}$ ←tune($tuning\_corpus_{ruen}$)
        $M_{ruen}$ ←getBestEns($models_{ruen}$)

        $bktr$ ←translate($mono_{ru}, M_{ruen}$)
        $dist$ ←distill($bitext_{enru}, M_{enru}$)
        $corpus = bktr + dist + bitext_{enru}$
        $model_{enru}$ ←train($corpus$)
        $model_{enru}$ ←tune($tuning\_corpus_{enru}$)
        $M_{enru}$ ←getBestEns($models_{enru}$)

    end while
end procedure
```

2.3 Domain adaptation

WMT 2022, for the first time, allowed the usage of pre-trained masked language models (MLM; exclusively in BERT-based architecture). We leveraged them to extract domain-specific subsets of mono and parallel corpora to fine-tune our NMT models in two chosen domains: *legal* and *medical*. We divide our approach into three steps: 1) Rule-based seed extraction, 2) Iterative Classifier training 3) Domain corpora extraction. Domain adaptation was performed only for the EN↔RU language pair. Finally, we used corpora described in Section 2.3.4, to adapt to the competition test sets.

2.3.1 Rule-based seed extraction

Our work focuses on two non-news domains: *medical* and *law*. We prepared initial monolingual (EN) seed corpora based on handcrafted rules and manual filtering. The datasets were too small to perform fine-tuning of MLM; therefore, we added an intermediate step. We encoded the sentences using general-purpose BERT (Devlin et al., 2019) and applied a K Nearest Neighbours (KNN) classifier to filter the extended version of the initial corpora. The extended version was extracted using the same rules but without manual filtering.

2.3.2 Iterative Classifier training

We base our approach on tri-training (Zhou and Li, 2005; Ruder and Plank, 2018). Rule-based extracted seed serves as the training data, and the manually filtered examples are the test set. In contrast to the original tri-training, we enlarge our training dataset after training the three classifiers instead of continuously adding new examples during training (we call this an iteration). Due to time constraints, we performed two such iterations per domain. The classifiers are fine-tuned BERT models, yet domain-specific ones: Lee et al. (2019) for the *medical* domain and Chalkidis et al. (2020) – *legal*.

2.3.3 Domain corpora extraction

With the final ensemble of classifiers, we scored a subset of monolingual, English data (CommonCrawl) and parallel corpora, which was not used during the ensemble training. We raised a threshold for the classifiers to 0.9 and included a sentence to a domain using unanimous voting. The resulting monolingual/parallel corpora size is presented in Table 1.

	Domain	
Corpora	Medical	Legal
Monolingual	93.3	184.6
Parallel	5.1	135.5

Table 1: Size of extracted domain-specific corpora (in thousands)

2.3.4 Test set adaptation

The last step of domain adaptation was the WMT 2022 test set adaptation. Our main intention was to prepare a corpus based on sentences similar to those present in the competition test set. To achieve this goal, we used the KNN algorithm. The first step was creating a dataset consisting of sentence embeddings from the WMT 2022 test set and all constrained corpora. Embeddings were acquired using the BERT base model (cased) (Devlin et al., 2019). Afterwards, we applied the k nearest neighbours search. The parameters were selected empirically: the number of nearest neighbours was set to 20, and we chose the Euclidean distance metric. Finally, the candidates were picked by finding neighbours whose distance to a given sentence from WMT 2022 test set was lower than 1.2.

2.4 Dynamic ensemble weighting

For each given (expert-selected) collection of NMT models, two modes of ensemble translation were tested. In the standard mode, the entire test set is translated using the same "static" set of weights for ensemble components. Alternatively, we attempted to construct a regression model that would generate weights best suited to a given sentence type; we call this mode "dynamic". For this, we concatenated outputs from 3 BERT-based predictors, trained to classify sentences as belonging to legal, medical and news domain, respectively. The medical and legal predictors were as described in 2.3.2; the predictor for news domain was fine-tuned in the same way with the pretrained BERT model allenai/news-roberta-base. Because each predictor produced 6 strongly-correlated values, the resulting vectors underwent dimensionality reduction, before being passed as inputs to the weight regression model; the regression itself is a relatively simple affine transformation in the logit domain. We leveraged only English Bert models; therefore, in the RU→EN direction, we performed preliminary translation using some early ensemble and extracted the predictions from its English outputs; the Croatian task does not use weight optimization.

Because we could not perform a direct optimization of BLEU/chrF (Popović, 2015) with regard to ensemble weights (some sort of grid- or random-search would be possible, but was deemed too expensive), we settled on minimizing cross-entropy of reference translations. We experimented with two formulas for interpolation of probability distributions: in logarithmic-probability domain (more commonly found, e.g. in Marian), or in linear-probability domain.[1] However, because the minimization of cross-entropy in log-P domain will degenerate the ensemble to the single best model (it can be easily shown), we added the regularization parameter to optimization of this kind of ensembles. The regularization term penalizes the divergence from the uniform vector.

26k sentences were selected from the model training corpora as the training data, half of which was classified as news, the rest as legal, medical, or randomly sampled. Three validation sets were used: Flores, Newstest 2021 and training data held-out.

Static and dynamic weights were independently estimated using gradient-descent for a handful of different ensembles in each direction; the general observations on development sets were the following:

- for each of the directions, two different vectors/transforms seem to be optimal, depending on the development set (one for Flores, another for Newstest 2021 and held-outs)
- the impact of the interpolation model (log-P vs linear-domain) is moderate, usually with small advantage of log-P, except for EN→RU Flores where linear yields ca. +0.22 BLEU
- the impact of the dynamic weighting is minimal, giving 0.09 BLEU improvement on RU→EN direction, with 0.1–0.3 BLEU degradation on top EN→RU configurations.

For final submission, in RU→EN direction we used static ensembles as described; however, in EN→RU task we made a last-minute decision

[1] We added an in-house extension to Marian-NMT that implements this alternative ensemble interpolation (i.e. done in the linear-probability domain); a patch that facilitates running ensemble translations with a weight vector different for every sentence was also implemented.

to scrap automatically-derived weights and used expert-crafted ensembles (obviously, also static).

We conjecture that the reason for the limited benefits from the above experiments lies in the indirect optimization of BLEU through cross-entropy, as well as – in the dynamic approach – in small actual distinctiveness of domain-specific data.

3 English-Russian

All corpora were preprocessed by removing sentences of inappropriate languages, normalizing punctuation, replacing all Russian letters ё (yo) with е (ye), removing duplicate sentences.

3.1 Parallel corpora

During the training, we used all accessible English-Russian parallel data except UEDIN back-translated news corpus. This corpus was used only during the first training iteration before generating any new back-translated data. Later it was excluded from training because it was worsening the results. We filtered sentence pairs where the length ratio between source and target sentences exceeded 1.6. Paracrawl (Bañón et al., 2020) paragraphs consisting of more than one sentence were split into single sentences and appended to the original dataset.

We used our in-house rule-based filtering, but we did not detect improvement but worsened quality over not-filtered data. Similarly, inferior results were obtained by applying Cross-Entropy Filtering (Junczys-Dowmunt, 2018). Therefore, we used unfiltered data during most of the training process.

3.2 Monolingual corpora

We used the monolingual corpora in two ways: to train language models and to augment the parallel data with back-translated data. Back-translation (Sennrich et al., 2016) is a commonly used technique for improving machine translation, especially for low-resource languages (Edunov et al., 2018).

We chose three different sources of monolingual corpora and preprocessed them similarly to parallel data (with minimal preprocessing). The used corpora are:

- News crawl
- CommonCrawl
- News-CommonCrawl

All corpora were filtered by a language model trained on the same corpus leaving only sentences with a likelihood larger than 1e-5. Due to the poor quality of CommonCrawl, we used only lines/paragraphs containing three or more sentences, which we split into single sentences.

News-CommonCrawl is the same filtered CommonCrawl but additionally filtered by a fastText[2] model trained on 100k news sentences from News crawl and 100k sentences from CommonCrawl. Using this model, we selected sentences classified by fastText as news (Joulin et al., 2017).

During all training iterations, except the first, we back-translated monolingual data using the best ensembles of currently trained models. We used clean back-translation as well as noised (Edunov et al., 2018) and tagged back-translation (Caswell et al., 2019). We applied gumbel noise for noised back-translation, as implemented in Marian, changing the epsilon value from default 1e-5 to 1e-3.

3.3 Teacher-Student Knowledge Distillation

Distilled corpora were prepared by translating parallel corpora using best ensembles in the direction of training with a beam equal to eight and selecting two translations most similar to the original translation. Such corpus was added to the parallel corpus expanding it three times.

3.4 Tuning corpora - FLORES

Despite poor results of standard filtering, we experimented with modified filtering versions during further iterations. We finally found the following filtering by marian-scorer that applied to parallel corpora improved results in some of the final training iterations.

- Language model filtering - Using a language model trained on a monolingual corpus we filtered utterances for which the normalized likelihood of the target side was higher than 1e-5.
- Backward cross-entropy filtering - Using the backward translation model, we filtered only sentence pairs where target to source translation normalized likelihood was larger than 1e-2.

The filtering described above was not applied to the Wikititles corpus.

[2] https://fasttext.cc

3.5 Tuning corpora - NEWS

Models adapted for news were finetuned by two consecutive tuning iterations using the following corpora:

1. Paracrawl and News Commentary
2. News Commentary and all Newstests from WMT2012-20

3.6 Contextual corpus and decoding

The corpus used for contextual training translation was built of two parts:

- Parallel utterances from News Commentary containing 2-4 subsequent sentences.
- Sequence of 2-4 adjacent sentences from one paragraph of CommonCrawl monolingual corpus, back-translated sentence by sentence. The back-translated part was tagged.

During decoding, we translated a sentence four times:

- without a context
- with one preceding sentence
- with two preceding sentences
- with two preceding and one following sentence

From the four above translations, we chose the translation most similar to 3 others using Levenshtein distance (Levenshtein, 1965) as a similarity metric.

4 English-Croatian

We applied similar preprocessing as for Russian language. Additionally to all available EN-HR corpora from OPUS (Tiedemann, 2012) we added all available data for Serbian language to the training. We used custom validation set based on TED for first iterations and WMT22 dev set for last two iterations. We added directional tokens in front of each sentence that allowed to differentiate between Croatian and Serbian translation.

For back-translation we used news mono corpora and source language from all EN-HR parallel corpora as well. Additionally to the back-translated corpora we added EN-HR parallel data. We performed two iterations of back-translation. After training of first iteration with back-translated data we fine-tuned the model on all parallel EN-HR data. After training of the second iteration we fine-tuned the model on CCMatrix corpus (Schwenk et al., 2021). The back-translation was noised with gumbel noise.

After the above we started to apply knowledge distillation and fine-tuning the model on distilled data. We did only 2 iterations of distillation. First distillation was done on CCMatrix corpus and second on tuning corpus (created from DGT, QED, TedTalks, EuroPat, SETIMES, hrenWaC, TED2020 corpora). We experimented with different learning rates in order to find the best performing model after this step. Finally, we made an ensemble out of the best-performing models. Additionally, we found that a normalization value of 0.5 results in a better score.

5 Results

Results of training iterations for English to Russian are presented in Table 2. Table 3 presents results for the Russian to English direction. Finally, Table 4 presents results for the English to Croatian task. Abbreviations mean:

- BTN - noised back-translation
- BTT - tagged BT
- BTTN - tagged noised BT
- KD - training with distilled parallel corpus
- news / cc / ncc - back-translated corpus
  - News crawl
  - CommonCrawl
  - News-Commoncrawl

First iteration was trained using only constrained parallel corpora provided by organizers. Next iterations were trained on mixed parallel corpora combined with back-translated monolingual data (BT). Further iterations used also distilled forward translations (KD).

Tuning with domain adaptation corpora has improved slightly (0.1-0.2) some of single models but gave no noticeable improvement on final score of ensembles.

6 Conclusions

We confirmed that iterative knowledge distillation combined with iterative back-translation is sufficient to prepare high-quality translation models.

Iter	Corpus	Flores devtest	Newstest 2021
0	Parallel – baseline	30.1	26.8
1	BTN-news	**32.5**	**28.8**
	BTN-news, filtered bitext	31.9	28.3
2	KD BTN-news	33.7	29.5
	KD BTT-news	**34.0**	**29.6**
	KD BTT-cc	33.9	28.9
3	KD BTT-news	**34.1**	29.4
	KD BTT-news, tuned news	33.9	**29.9**
4	KD BTN-news	33.0	29.4
	KD BTTN-news	33.7	29.2
	KD BTT-news	34.0	29.6
	KD BTT-news, tuned news	33.6	**30.1**
	KD BTT-news + context	33.8	29.7
	KD BTT-cc	34.4	28.9
	KD BTT-cc + context	**34.5**	28.8
Best ensemble flores - SRPOL submission		34.8	30.7
Best constrained WMT2021			**29.3**

Table 2: Iterations and results of training for EN→RU direction.

Iter	Corpus	Flores devtest	Newstest 2021
0	Parallel – baseline	35.6	35.5
1	BTN-news	**37.0**	36.5
	KD + BTN-news	36.4	**37.9**
2	BTN-news	**36.6**	**37.1**
	BTN-news, filtered bitext	36.2	36.7
3	BTN-news	37.4	37.6
	BTN-ncc	**38.1**	36.7
	KD + BTN-ncc	37.8	37.9
	KD + BTTN-ncc	37.4	**39.0**
4	KD + BTT-ncc	37.0	38.8
	KD + BTN-ncc	38.0	38.1
	KD + BTTN-ncc	37.4	38.5
	KD + BTT-ncc tuned news	37.0	**40.2**
	KD + BTN-ncc tuned news	37.8	39.8
	KD + BTN-ncc + context	**38.1**	38.2
	KD + BTN-ncc + context tuned news	37.6	39.7
Best ensemble flores - SRPOL submission		**38.9**	40.8
Best ensemble news		38.3	**41.6**
Best constrained WMT2021			**41.8**

Table 3: Iterations and results of training for RU→EN direction.

Iter	Corpus	Flores devtest	WMT 22 devtest
0	Parallel – baseline	31.9	32.1
1	BTN	32.7	32.0
2	BTN	32.8	32.2
3	KD	33.5	33.4
4	KD	33.7	33.3
Best ensemble + normalization - SRPOL submission		33.6	33.7

Table 4: Iterations and results of training for EN→HR direction.

This method gives excellent results on low-resource and mid-resource languages. During the WMT 2022 General MT task, we reached one of the best results among constrained systems.

In our work, we compared different methods of back-translation: clean, noised, and tagged. Mostly, the tagged back-translation achieved the best results, but for some training iterations, noised back-translation's results were on-par or better.

We compared different sources of monolingual data used for back-translation: CommonCrawl and News crawl. The comparison suggests that the choice of the monolingual corpus has a significant influence on final results.

Our exploration of different filtering methods suggests that while using pre-filtered data (as provided in WMT 2022), it is sufficient to filter only target data, leaving source data unfiltered.

We presented a simple and effective method of adding contextual data to the training corpus, which gave a noticeable improvement.

We investigated a new method of dynamic ensemble weighting, but the results show no improvement over other methods.

Limitations

In our work we touched on a few aspects but did not have time to address them in more detail.

The research showed that tagged back-translation generally gives better results than other back-translation methods, but not always. It may be worth to investigate more deeply methods of different noising, different noise level and how it synergies on various parallel and monolingual corpora.

Almost all our training iterations were performed on very similar default transformer-big configurations. We haven't tested other configurations, larger or deeper models, different training parameters, what can improve the results.

We introduced very simple contextual translation method which can be improved in many ways.

We gained best results filtering data only on target size, leaving source data unfiltered. This issue looks worth to be investigated.

Ethics Statement

During our work we followed all the rules of ACL Ethics Policy

All our efforts aimed at conducting research for the benefit of society and to human well-being. During our research we used only fair and honest methods. We didn't hide any information needed to repeat our results. We didn't use any resources out of the data provided by organizers in constrained path.

Acknowledgements

We would like to thank all the people who helped us during the research with their consultations and resources. Special thanks to SRPOL's Linguistic Department, SRPOL's Management Staff, Samsung Research HQ in Suwon, Korea.

References

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Vladimir I. Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady*, 10:707–710.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Sebastian Ruder and Barbara Plank. 2018. Strong baselines for neural semi-supervised learning under domain shift. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1044–1054, Melbourne, Australia. Association for Computational Linguistics.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and*

Evaluation (LREC'12), pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Shuhan Zhou, Tao Zhou, Binghao Wei, Yingfeng Luo, Yongyu Mu, Zefan Zhou, Chenglong Wang, Xuanjun Zhou, Chuanhao Lv, Yi Jing, Laohu Wang, Jingnan Zhang, Canan Huang, Zhongxiang Yan, Chi Hu, Bei Li, Tong Xiao, and Jingbo Zhu. 2021. The NiuTrans machine translation systems for WMT21. In *Proceedings of the Sixth Conference on Machine Translation*, pages 265–272, Online. Association for Computational Linguistics.

Zhi-Hua Zhou and Ming Li. 2005. Tri-training: exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1529–1541.

Tencent AI Lab - Shanghai Jiao Tong University Low-Resource Translation System for the WMT22 Translation Task

Zhiwei He[*]
Shanghai Jiao Tong University
zwhe.cs@sjtu.edu.cn

Xing Wang[†]
Tencent AI Lab
brightxwang@tencent.com

Zhaopeng Tu
Tencent AI Lab
zptu@tencent.com

Shuming Shi
Tencent AI Lab
shumingshi@tencent.com

Rui Wang
Shanghai Jiao Tong University
wangrui12@sjtu.edu.cn

Abstract

This paper describes Tencent AI Lab - Shanghai Jiao Tong University (TAL-SJTU) Low-Resource Translation systems for the WMT22 shared task. We participate in the general translation task on English⇔Livonian. Our system is based on M2M100 (Fan et al., 2021) with novel techniques that adapt it to the target language pair. (1) Cross-model word embedding alignment: inspired by cross-lingual word embedding alignment, we successfully transfer a pre-trained word embedding to M2M100, enabling it to support Livonian. (2) Gradual adaptation strategy: we exploit Estonian and Latvian as auxiliary languages for many-to-many translation training and then adapt to English-Livonian. (3) Data augmentation: to enlarge the parallel data for English-Livonian, we construct pseudo-parallel data with Estonian and Latvian as pivot languages. (4) Fine-tuning: to make the most of all available data, we fine-tune the model with the validation set and online back-translation, further boosting the performance. In model evaluation: (1) We find that previous work (Rikters et al., 2022) underestimated the translation performance of Livonian due to inconsistent Unicode normalization, which may cause a discrepancy of up to 14.9 BLEU score. (2) In addition to the standard validation set, we also employ round-trip BLEU to evaluate the models, which we find more appropriate for this task. Finally, our unconstrained system achieves BLEU scores of 17.0 and 30.4 for English to/from Livonian.[1]

1 Introduction

This paper introduces our submissions to the WMT22 general machine translation task. Last year, Tencent AI Lab participated in two translation tasks: News (Wang et al., 2021a) and Biomedical translation (Wang et al., 2021b). This year, we participate in English⇔Livonian (En⇔Liv), a very low-resource and distant language pair. Considering the scarcity of parallel En-Liv corpus, we only participate in the unconstrained evaluation.

We use M2M100 1.2B[2] (Fan et al., 2021) as the pre-trained model which is a massive multilingual translation model that supports any pair of 100 languages[3] and shows promising performance for low-resource translation. To adapt it to En-Liv, the first thing to do is enabling it to support Liv. A common approach is to expand the vocabulary and the word embedding matrix to contain the extra tokens. However, the incoming embeddings must be randomly initialized (Garcia et al., 2021; Bapna et al., 2022), which leads to inconsistency with the original embeddings and increases training difficulty. Fortunately, Rikters et al. (2022) has released a translation model for En-Liv called Liv4ever-MT[4]. Inspired by supervised cross-lingual word embedding alignment (Lample et al., 2018b), we propose cross-model word embedding alignment (CMEA) that learns a linear transformation between the embedding matrices of two models. Therefore, the incoming embeddings can be extracted from Liv4ever-MT and transformed to M2M100's word embedding space rather than random initialization.

In terms of model training, we adopt a gradual adaptation strategy. The overall training process is shown in Figure 1. Following Rikters et al. (2022), we also use Estonian (Et) and Latvian (Lv) as auxiliary languages. Liv has been influenced by Et and Lv for centuries. There are about 800 Et loanwords and 2,000 Lv loanwords in Liv (Décsy, 1965). Therefore, we first add Et and Lv for many-to-many translation training, resulting in a 4-lingual

[*] Work was done when Zhiwei He was interning at Tencent AI Lab.

[†] Xing Wang is the corresponding author.

[1] Code, data, and trained models are available at https://github.com/zwhe99/WMT22-En-Liv.

[2] https://github.com/facebookresearch/fairseq/tree/main/examples/m2m_100

[3] M2M100 supports English, Latvian and Estonian.

[4] https://huggingface.co/tartuNLP/liv4ever-mt

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 260 - 267
December 7-8, 2022 ©2022 Association for Computational Linguistics

translation model. We then augment the En-Liv data with forward and backward translations using Et and Lv as the pivot languages. Finally, we combine all the authentic and synthetic data to retrain the model, followed by a few steps of fine-tuning with the validation set and online back-translation.

In terms of model evaluation, we find that the data set provided by Rikters et al. (2022) suffers from inconsistent Unicode normalization. This inconsistency is reflected in using two or more encodings for the same character, which leads to inconsistent encoding between model hypothesis and reference[5] and thus inaccurate evaluation. In our experiments, normalizing the character encoding can bring an average improvement of +2.5 BLEU on the liv4ever[6] test set (see appendix A) and up to +14.9 BLEU on a subset from a specific source. In addition to the standard validation set, we also employ round-trip BLEU to evaluate our models, which is an effective unsupervised criterion (Lample et al., 2018a) and reduces the demand for the parallel corpus. Zhuo et al. (2022) have found that in the scope of neural machine translation, round-trip translation quality correlates consistently with forward translation quality. We consider round-trip BLEU a better evaluation method for this task. The reasons for this are threefold: more data, more general domain, and the same original language as the WMT22 En-Liv test set.

This paper is structured as follows: Section 2 describes the data statistics and processing methods. Then we present our evaluation methods in Section 3. Our translation system and ablation study are detailed in Section 4, followed by the final results. Finally, we conclude the paper in Section 5.

2 Data and Processing

2.1 Overview

Statistics Table 1 lists statistics of the parallel and monolingual data we used. We collect parallel data for any pair in {En, Liv, Et, Lv} and collect monolingual data for En and Liv.

Data	Lang	# Sent.	
		Raw	Filter
Parallel Data	En-Liv	1.2K	1.1K
	En-Et	40.3M	20.7M
	En-Lv	27.2M	11.3M
	Liv-Et	14.8K	14.8K
	Liv-Lv	12.4K	12.2K
	Et-Lv	10.7M	7.0M
Monolingual Data	En	325.6M	281.3M
	Liv	138.2K	50.2K

Table 1: Statistics of parallel and monolingual data. We report the number of sentences before and after filtering.

Data Source The parallel data is mainly all available corpora from OPUS[7]. Due to the scarcity of data, we include liv4ever-dev in training data and use liv4ever-test as the validation set. For En-Et and En-Lv, we augment them with the parallel data from WMT18 and WMT17, respectively. For En-Liv, En-Lv and Liv-Lv, we collected additional parallel data from Facebook posts of the Livonian Institute and Livones.net[8]. The monolingual En is News Crawl 2007-2021. The monolingual Liv combines all Liv from parallel data and monolingual data from liv4ever[6].

2.2 Pre-processing

To obtain higher quality training data, we employ a series of data cleaning using Moses toolkit[9] and our scripts[10]. We process parallel data as follows:

- Replace Unicode punctuation, normalize punctuation and remove non-printing characters
- Language identification and filtering
- Remove instances with too much punctuation
- Remove instances with identical source and target sentences
- Remove instances containing URLs
- Remove instances appearing in evaluation data
- Remove instances with more than 175 tokens or length ratio over 1.5

The liv4ever corpus has a small amount of data, and the existing tools may not support Liv well. Therefore, for the liv4ever corpus, we don't apply punctuation processing or language and length ratio filtering. For the monolingual data, we use the same cleaning steps as parallel data except for

[5] SentencePiece does uniform normalization by default. Therefore, the character encoding in the model hypothesis is uniform but may not be consistent with the reference.

[6] https://opus.nlpl.eu/liv4ever-v1.php

[7] https://opus.nlpl.eu/

[8] The numbers of additional sentences collected from Facebook are En-Liv: 54, En-Lv: 61 and Liv-Lv: 61.

[9] https://github.com/moses-smt/mosesdecoder

[10] https://github.com/zwhe99/corpus-tools

identical source-target filtering and length ratio filtering.

After cleaning the data, we apply SentencePiece[11] encoding using the trained model from Liv4ever-MT[4]. We also reuse their vocabulary that shared by all languages.

2.3 Evaluation Data

We regard the liv4ever-test as the validation set, which is a multi-way data set for {En, Liv, Et, Lv} containing 855 unique sentences. Besides, for En⇔Liv evaluation, we collect monolingual English from the source of WMT22 English-German (En-De) test set to compute round-trip BLEU (En⇒Liv⇒En).

3 Model Evaluation

This section describes our methods for model evaluation. Specifically, we explain the Unicode inconsistency problem in the liv4ever data set and the resulting underestimation of model performance. In addition, we introduce round-trip BLEU as the more appropriate way for this competition.

3.1 Unicode inconsistency problem

Rikters et al. (2022) collected the liv4ever data set and built Liv4ever-MT, the first machine translation model for Livonian. We find that the liv4ever data set does not use consistent Unicode normalization, resulting in different encodings for the same character. This did not lead to any training problem in Rikters et al. (2022) because SentencePiece does NFKC[12] normalization by default. However, when computing SacreBLEU[13], the encoding of model output and the reference will be inconsistent, resulting in inaccurate evaluation.

We re-evaluate the performance of Liv4ever-MT before and after normalizing the encoding of references to NFKC. Table 2 shows the SacreBLEU results[14] on the entire test set and a subset from Satversme. Before normalization, our results are very close to those reported in Rikters et al. (2022), while after normalization, the BLEU score improves considerably. In particular, the difference in BLEU score is up to 14.9 on the Lv⇒Liv of the Satversme subset. Therefore, we report SacreBLEU after normalization in the following.

	En-Liv		**Et-Liv**		**Lv-Liv**	
	⇒	⇐	⇒	⇐	⇒	⇐
			All			
Liv4ever-MT (Rikters et al.)	11.0	19.0	16.5	23.1	17.7	25.2
Our Eval.	10.9	18.9	16.6	22.9	17.7	24.9
+ Norm. Ref.	**14.3**	**19.3**	**20.5**	**24.4**	**22.3**	**29.3**
			Subset (Satversme)			
Liv4ever-MT (Rikters et al.)	7.7	24.5	-	-	-	-
Our Eval.	7.6	24.7	7.2	18.7	9.2	19.4
+ Norm. Ref.	**18.2**	**25.8**	**19.9**	**23.7**	**24.2**	**33.6**

Table 2: BLEU scores of Liv4ever-MT on liv4ever-test. **Liv4ever-MT** (Rikters et al.): copied from Rikters et al. (2022). **Our Eval.**: We use the released Liv4ever-MT to generate translation outputs and re-evaluate them with the original references, which shows similar results compared with Rikters et al. (2022). **+ Norm. Ref.**: re-evaluation after normalizing the encoding of references to NFKC. See Appendix A for all language pairs.

3.2 Round-trip BLEU

We collect monolingual English from the source of WMT22 English-German (En-De) test set and conduct two steps translation: En⇒Liv⇒En. The round-trip BLEU score can be obtained by comparing the original input with the model output English. We regard it a better way to evaluate En⇔Liv performance for this task considering three aspects: (1) En-De test set has 20683 sentences, much more than the liv4ever-test. (2) It may contain more general domain data, while the liv4ever-test is relatively restricted due to the low-resource limitation. (3) The original language used in computing the round-trip BLEU is the same as the WMT22 En-Liv test set (both English-original).

4 System and Ablation Study

In this section, we describe our system in this competition and provide a comprehensive ablation study of the key components.

4.1 System Overview

We depict the overview of our system in Figure 1, which can be divided into five steps:

1. **Cross-model word embedding alignment**: transfer the word embeddings of Liv4ever-MT to M2M100, enabling it to support Livonian.

[11] https://github.com/google/sentencepiece

[12] https://unicode.org/reports/tr15/

[13] https://github.com/mjpost/sacrebleu

[14] nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.0

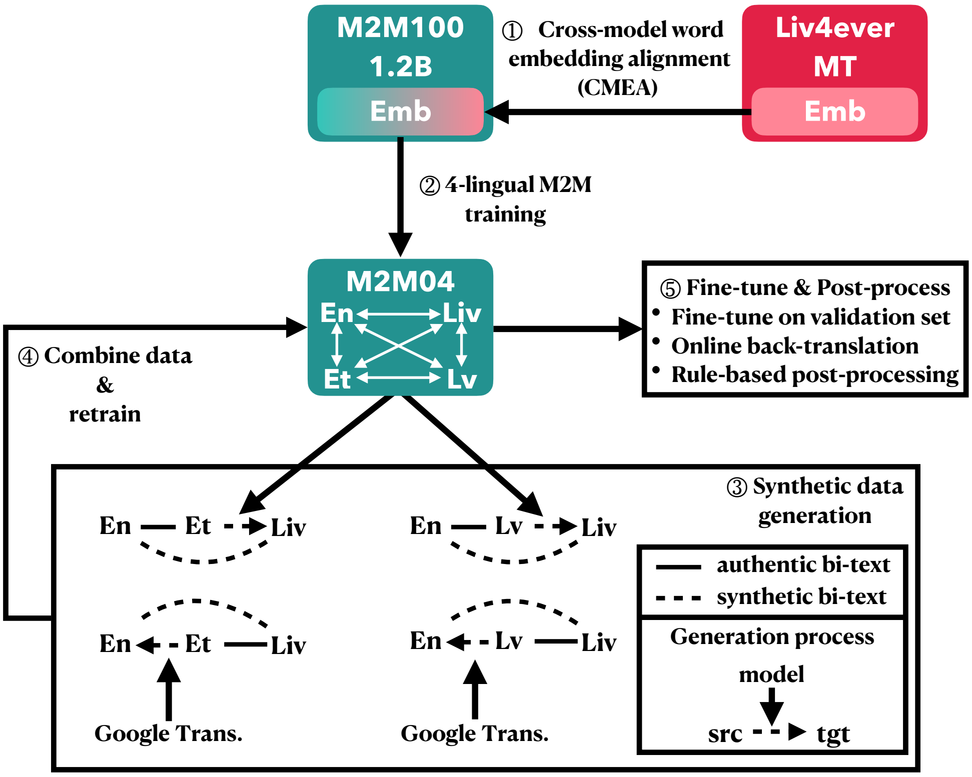


Figure 1: The training process of our translation system.

2. **4-lingual M2M training**: many-to-many translation training for all language pairs in {En, Liv, Et, Lv}, using only parallel data.
3. **Synthetic data generation**: generate synthetic bi-text for En-Liv, using Et and Lv as pivot languages.
4. **Combine data and retrain**: combine all the authentic and synthetic bi-text and retrain the model following step 2.
5. **Fine-tune & post-process**: fine-tune the model on En⇔Liv using the validation set and perform online back-translation using monolingual data. Finally, apply rule-based post-processing to the model output.

4.2 Cross-model Word Embedding Alignment

M2M100 1.2B does not support Livonian. Therefore, we used Liv4ever-MT's SentencePiece model and vocabulary to process all the data. For M2M100, the embeddings of new coming words can be randomly initialized. However, randomly initialized word embeddings and the pre-trained models may not be compatible. Inspired by supervised cross-lingual word embedding alignment (Lample et al., 2018b), we propose cross-model word embedding alignment (CMEA) to transform the trained word embeddings of Liv4ever-MT into M2M100, avoiding random initialization.

CMEA We denote Liv4ever-MT and M2M100 model by l and m. Their corresponding vocabularies and embedding matrices are d_l, d_m and $\mathbf{X}^l$, $\mathbf{X}^m$. Table 3 shows the statistics of the vocabularies. Let $\mathbf{X}^f$ be the final embedding matrix we expected. We adopt d_l as the final vocabulary, which can be divided into two parts:

$\lvert d_l \rvert$	$\lvert d_m \rvert$	$\lvert d_l \cap d_m \rvert$	$\lvert d_l \cap d_m \rvert / \lvert d_l \rvert$
47972	128108	11410	23.8%

Table 3: Statistics of Liv4ever-MT (d_l) and M2M100 (d_m) vocabularies.

$$d_l = (d_l \cap d_m) \cup (d_l - d_m). \qquad (1)$$

For the overlapped part $d_l \cap d_m$, $\mathbf{X}^f$ can reuse the embedding from $\mathbf{X}^m$:

$$\mathbf{X}^f_{d_l \cap d_m} = \mathbf{X}^m_{d_l \cap d_m}. \qquad (2)$$

For the rest part $d_l - d_m$, we first find a liner transformation $\mathbf{W}$ between two embedding spaces such that:

$$\mathbf{W}^* = \underset{\mathbf{W}}{\arg\min} \|\mathbf{W}\mathbf{X}^l_{d_l \cap d_m} - \mathbf{X}^m_{d_l \cap d_m}\|_F \qquad (3)$$
$$\text{s.t.}\ \mathbf{W}^T\mathbf{W} = \mathbf{I}.$$

According to Everson (1998),

$$\mathbf{W}^* = \mathbf{U}\mathbf{V}^T, \qquad (4)$$
$$\text{with } \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \text{SVD}\left(\mathbf{X}^m_{d_l \cap d_m} {\mathbf{X}^l_{d_l \cap d_m}}^T\right).$$

Then the word embeddings can be initialized as:

$$\mathbf{X}^f_{d_l - d_m} = \mathbf{W}^*\mathbf{X}^l_{d_l - d_m}. \qquad (5)$$

Experiment To investigate the effect of CMEA, we conducted **4-lingual M2M training** with different sampling temperature (Aharoni et al., 2019; Tang et al., 2021). Table 4 shows the BLEU scores on the validation set. We have the following observations:

- M2M04 outperforms Liv4ever-MT by a large margin owing to the larger model size, more training data and the pre-trained parameters.
- On most language pairs, our proposed CMEA initialization significantly improves translation performance compared to random initialization of new coming embeddings.
- Temperature set to 5 with CMEA initialization achieves the best overall results. Therefore, we used this model in **synthetic data generation**.

	En-Liv		**Et-Liv**		**Lv-Liv**	
	⇒	⇐	⇒	⇐	⇒	⇐
Liv4ever-MT Rikters et al.	14.3	19.3	20.5	24.4	22.3	29.3
M2M04 (T=5)	21.1	27.7	25.3	29.2	26.8	36.6
+ CMEA	**23.0**	**28.4**	**27.2**	**30.7**	**28.5**	**37.6**
M2M04 (T=10)	**21.3**	26.6	25.5	27.7	26.3	34.6
+ CMEA	21.1	**27.1**	**26.0**	**29.6**	**27.5**	**36.3**
M2M04 (T=20)	21.9	26.7	**26.5**	**29.8**	27.3	**36.5**
+ CMEA	**22.1**	**27.4**	25.8	27.9	**27.9**	33.8

Table 4: Experimental results of 4-lingual M2M training. We denote M2M04 as the 4-lingual translation model. 'T' represents the sampling temperature.

4.3 Synthetic Data Generation

Data agumentation (Sennrich et al., 2016; Jiao et al., 2020, 2022, 2021; He et al., 2022) is a widely used technique to boost the performance of neural machine translation. To augment the parallel data for En-Liv, we adopt both forward and backward translation to generate synthetic bi-text for En-Liv. Figure 1 (below) illustrates the process of synthetic data generation.

Considering the performances of Et/Lv⇒Liv are much better than En⇒Liv (see Table 4), we use Et and Lv as pivot languages to generate Liv instead of directly generating from En. Taking Et as the pivot language, given authentic En-Et bi-text, we use the best model in Table 4 to translate the Et into Liv, thus forming the synthetic En-Liv which is En-original. Conversely, given authentic Et-Liv, we translate Et into En using Google Translate, forming the synthetic En-Liv which is Liv-original. For Lv as the pivot language, we repeat the same steps. Table 5 lists statistics of the synthetic En-Liv data after filtering.

Data Type	**Pivot Language**	
	Et	Lv
En-original	20.5M	11.2M
Liv-original	14.2K	11.6K

Table 5: The number of sentences of generated synthetic data after filtering, which is divided into four categories based on the original language and the pivot language.

Experiment We combine the authentic and synthetic bi-text and retrain the 4-lingual model. The sampling temperature is set to 0 here to avoid downsampling for En-Liv. When using only En-original or Liv-original synthetic data, we control the sampling frequency of the different language pairs to be consistent with using the full data. Table 6 shows the BLEU scores on the multi-way validation set. We also report the round-trip BLEU on the monolingual En from the source of WMT22 En-De test set, which is En-original. Unexpectedly, original-language greatly affects the model performance and causes inconsistent results between different evaluation methods:

- En-original synthetic data remarkably degrades model performance on the validation set but significantly increases the round-trip BLEU.
- Liv-original synthetic data slightly reduces the performance on the validation set but moderately increases the round-trip BLEU.

- When using both kinds of data, the best round-trip BLEU is achieved. However, the performance on the validation set is still worse than the baseline.

	Valid (multi-way)		Round-Trip (En-original)
	En⇒Liv	Liv⇒En	
M2M04 (T=5) +CMEA	23.0	28.4	23.4
Add synthetic data and retrain			
En-original	17.2	17.5	30.7
Liv-original	21.5	27.4	25.8
Both	17.0	19.3	32.7

Table 6: Translation performance after adding the synthetic data and retraining the model.

As described in Section 3.2, we consider round-trip BLEU the more appropriate evaluation in this competition due to more data, more general domain, and the same original language as the WMT22 En-Liv test set. Therefore, we used both kinds of synthetic data in our submissions.

4.4 Fine-tuning & Post-processing

Fine-tuning To further exploit the bilingual and monolingual data, we fine-tuned the model on the En⇔Liv validation set for 500 steps jointly with online back-translation on monolingual data.

Post-processing We apply the following rule-based post-processing:

- Apply NFC normalization.
- Replace all the `httpshttp` with `https://`.
- Replace <unk> with empty string.
- When a comma appears between two digits, replace it with a decimal point (only for Liv).
- Regenerate the sentences that detected as repetition with no-repeat constraint[15] (only for Liv).

Final results Table 7 shows the test set performance and round-trip BLEU after fine-tuning and post-processing. As seen, fine-tuning significantly improves model performance on both test set and round-trip BLEU. Post-processing further boosts the performance on the test set.

	Test Set En-Liv		Round-Trip BLEU
	⇒	⇐	
Before fine-tuning	15.8	29.4	32.7
+Fine-tuning	16.3	30.1	37.1
+Post-proc.	17.0	30.4	37.1

Table 7: Translation performance after fine-tuning and post-processing.

5 Conclusion

This paper presents the Tencent AI Lab - Shanghai Jiao Tong University (TAL-SJTU) Low-Resource Translation systems for the WMT22 shared task. We start from the M2M100 1.2B model and investigate techniques to adapt it to English⇔Livonian. We propose cross-model word embedding alignment that transfer the embeddings of Liv4ever-MT to M2M100, enabling it to support Livonian. Then, Estonian and Latvian are involved in model training and synthetic data generation as auxiliary and pivot languages. We further fine-tune the model with validation set and online back-translation followed by rule-based post-processing. In model evaluation, we correct the inaccurate evaluation of Livonian due to inconsistent Unicode normalization and use round-trip BLEU as an alternative to the standard validation set.

Acknowledgements

Zhiwei He and Rui Wang are with MT-Lab, Department of Computer Science and Engineering, School of Electronic Information and Electrical Engineering, and also with the MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200204, China. Rui is supported by General Program of National Natural Science Foundation of China (6217020129), Shanghai Pujiang Program (21PJ1406800), and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102). Zhiwei is supported by CCF-Tencent Open Fund (RAGR20210119).

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computa-*

[15] We use `--no-repeat-ngram-size 2` in fairseq-generate.

tional Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics.

Ankur Bapna, Isaac Caswell, Julia Kreutzer, Orhan Firat, Daan van Esch, Aditya Siddhant, Mengmeng Niu, Pallavi Baljekar, Xavier Garcia, Wolfgang Macherey, et al. 2022. Building machine translation systems for the next thousand languages. *arXiv preprint arXiv:2205.03983*.

G. Décsy. 1965. *Einführung in die finnisch-ugrische Sprachwissenschaft*. O. Harrassowitz.

Richard Everson. 1998. Orthogonal, but not orthonormal, procrustes problems. *Advances in computational Mathematics*, 3(4).

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*.

Xavier Garcia, Noah Constant, Ankur Parikh, and Orhan Firat. 2021. Towards continual learning for multilingual machine translation via vocabulary substitution. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Zhiwei He, Xing Wang, Rui Wang, Shuming Shi, and Zhaopeng Tu. 2022. Bridging the data gap between training and inference for unsupervised neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Wenxiang Jiao, Xing Wang, Shilin He, Irwin King, Michael Lyu, and Zhaopeng Tu. 2020. Data rejuvenation: Exploiting inactive training examples for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Wenxiang Jiao, Xing Wang, Shilin He, Zhaopeng Tu, Irwin King, and Michael R Lyu. 2022. Exploiting inactive examples for natural language generation with data rejuvenation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30.

Wenxiang Jiao, Xing Wang, Zhaopeng Tu, Shuming Shi, Michael Lyu, and Irwin King. 2021. Self-training sampling with monolingual data uncertainty for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018b. Word translation without parallel data. In *International Conference on Learning Representations*.

Matīss Rikters, Marili Tomingas, Tuuli Tuisk, Valts Ernštreits, and Mark Fishel. 2022. Machine translation for Livonian: Catering to 20 speakers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics.

Longyue Wang, Mu Li, Fangxu Liu, Shuming Shi, Zhaopeng Tu, Xing Wang, Shuangzhi Wu, Jiali Zeng, and Wen Zhang. 2021a. Tencent translation system for the wmt21 news translation task. In *Proceedings of the Sixth Conference on Machine Translation*.

Xing Wang, Zhaopeng Tu, and Shuming Shi. 2021b. Tencent ai lab machine translation systems for the wmt21 biomedical translation task. In *Proceedings of the Sixth Conference on Machine Translation*.

Terry Yue Zhuo, Qiongkai Xu, Xuanli He, and Trevor Cohn. 2022. Rethinking round-trip translation for automatic machine translation evaluation. *arXiv preprint arXiv:2209.07351*.

	XX⇒En			**XX⇒Et**			**XX⇒Lv**			**XX⇒Liv**			**Avg.**
XX	**Et**	**Lv**	**Liv**	**En**	**Lv**	**Liv**	**En**	**Et**	**Liv**	**En**	**Et**	**Lv**	
	All												
Liv4ever-MT (Rikters et al.)	26.17	21.53	19.01	19.48	22.38	23.05	20.85	23.44	25.24	11.03	16.40	17.65	20.52
Our Eval.	25.90	17.94	18.90	19.28	22.31	22.86	20.20	23.31	24.88	10.90	16.62	17.69	20.07
+ Norm Ref.	**26.20**	**18.06**	**19.26**	**20.72**	**24.28**	**24.42**	**24.10**	**27.77**	**29.33**	**14.31**	**20.51**	**22.35**	**22.61**
	Subset (Satversme)												
Liv4ever-MT (Rikters et al.)	-	-	24.49	-	-	-	-	-	-	7.69	-	-	-
Our Eval.	27.50	19.77	24.68	16.69	20.22	18.68	16.05	15.10	19.38	7.58	7.18	9.23	16.83
+ Norm Ref.	**28.45**	**20.21**	**25.76**	**21.41**	**26.74**	**23.75**	**29.10**	**29.82**	**33.56**	**18.23**	**19.87**	**24.15**	**25.09**

Table 8: BLEU scores of Liv4ever-MT on liv4ever-test. **Liv4ever-MT** (Rikters et al.): copied from Rikters et al. (2022). **Our Eval.**: We use the released Liv4ever-MT to generate translation outputs and re-evaluate them with the original references, which shows similar results compared with Rikters et al. (2022). **+ Norm. Ref.**: re-evaluation after normalizing the encoding of references to NFKC.

A Re-evaluating Liv4ever-MT

Table 8 shows the results of re-evaluating Liv4ever-MT on all language pairs. Normalizing references to NFKC improves the average BLEU scores by +2.54 on the entire set and +8.26 on the Satversme subset. It is worth mentioning that liv4ever-test contains data from the following sources: Facebook, Livones.net, Dictionary, Trilium, Stalte, JEFUL and Satversme. However, there does not exist the Unicode inconsistency problem in the other sources except Satversme.

Lan-Bridge MT's Participation in the WMT 2022 General Translation Shared Task

Bing Han
Lan-Bridge / Sichuan (China)
hanbing@lan-bridge.com

Yangjian Wu
Lan-Bridge / Sichuan (China)
wuyangjian@lan-bridge.com

Gang Hu
Lan-Bridge / Sichuan (China)
hugang@lan-bridge.com

Qiulin Chen
Lan-Bridge / Sichuan (China)
chenqiulin@lan-bridge.com

Abstract

This paper describes Lan-Bridge Translation systems for the WMT 2022 General Translation shared task. We participate in 18 language directions: English to and from Czech, German, Ukrainian, Japanese, Russian, Chinese, English to Croatian, French to German, Yakut to and from Russian, and Ukrainian to and from Czech. We mainly focus on multilingual models to develop systems covering all these directions. In general, we apply data corpus filtering, scaling model size, sparse expert model (in particular, Transformer with adapters), large-scale backtranslation, and language model reranking techniques. Our system ranks first in 6 directions based on the automatic evaluation.

1 Introduction

Our Lan-Bridge MT team participate in the WMT 2022 General Translation shared task. As machine translation expands into more and more languages, multilingual machine translation has attracted more and more attention in both academia and industry. It can not only avoid training a separate model for each language pair but also transfer knowledge from high-resource languages to low-resource ones. Many systems such as Tran et al. (2021) submitted in previous years have proved this point and achieved a state of the art results in some language directions.

For data preprocessing, knowledge-based rules, language detection, and language model are involved to clean parallel data, monolingual data, and synthetic data (mainly from large-scale data mining and backtranslation). Punctuation normalization and BPE (byte pair encoding) (Sennrich et al., 2015) with subword regularization method (Provilkov et al., 2019) are applied for all languages. As for models, we fork Fairseq (Ott et al., 2019) as our development tool and use Transformer (Vaswani et al., 2017) as the main architecture. In addition, we follow Bapna et al. (2019) to extend Transformer by adding language-specific adapters to bridge the gap between different language pairs. Finally, we ensemble dense Transformer models and sparse adapter models, and the final result are re-ranked by language models. For English to and from Chinese, we develop a separate system. In addition to optimization techniques similar to multilingual models, We also use additional private data. And for Yakut to and from Russian, due to a smaller corpus, we simply apply fine-tuning and backtranslation on our multilingual models.

We win the first place in Russian ↔ Yakut, Russian → English, English → Croatian, Czech → English and Ukrainian → English based on BLEU (Papineni et al., 2002) score. [1]

2 System Overview

2.1 Data

Here we describe our base datasets, including bitext and monolingual data sources, and the preprocessing methods we apply to prepare these initial data sets to train our baseline models.

2.1.1 Bitext Data

We use all available bitext data from the shared task for all language pairs, besides, for English to and from Chinese, we add extra data from ai-challenger. For high-resource language pairs such as English to Chinese or English to German, which provides millions of high-quality bitext, we only choose those high-quality resources, and simply apply language identification using fasttext (Joulin et al., 2017) with an ID threshold of 0.8 and knowledge-based rules shows below as data process:

- Remove empty sentences
- De-escaping HTML characters

[1] This result is based on the submission website https://ocelot-wmt22.mteval.org/, not the official final result.

- Normalization of different languages of punctuation
- Normalization of spacing
- Remove sentences with repeated tokens, including single character that repeat more than four times, two characters that repeat more than three times, and more than three characters that repeat more than twice.
- Delete the corpus with inconsistent punctuation marks at the end of the original text and the translation
- Deletion of segments where source/target token ratio exceeds 1:3 (or 3:1)
- Deletion of segments longer than 150 tokens
- Deletion of segments shorter than 5 tokens
- Transfer traditional Chinese characters to simplified Chinese characters
- Delete corpus with misaligned number of parentheses
- Delete corpus with misaligned number of Arabic numerals
- Delete corpus with a proportion of non-native language characters exceeding 0.4

The normalization of spacing and punctuation is applied using Moses (Polykovskiy et al., 2020).

For medium- and low-resource language pairs, we incorporate additional sources of data from OPUS (Tiedemann, 2012), ccAligned (El-Kishky et al., 2020), and ccMatrix (Schwenk et al., 2019). All available data sources are utilized to train our models.

Due to the low-quality issue of corpora mentioned above, we add a few filter steps to make them usable. First, we try the word alignment method using fast_align (Dyer et al., 2013) to filter low-quality sentence pairs and keep top 80% for all directions ranked by alignment score. Then we use Fairseq to train the transformer multilingual language model for all languages, similar to Bei et al. (2019), the score is calculated as follows:

$$Score_{sentence} = PPL$$

$$Score_{combine} = \lambda * Score_{src} + (1 - \lambda) * Score_{tgt}$$

Language Pair	Data
cs-en	100M
de-en	250M
fr-de	20M
hr-en	70M
sah-ru	0.1M
uk-cs	6M
uk-en	20M
zh-en	50M
ru-en	80M
ja-en	20M

Table 1: Ultimate bitext training data

Language	Data
cs	64M
en	72M
de	63M
fr	79M
ja	81M
sah	0.2M
ru	70M
uk	5M
zh	10M
hr	14M

Table 2: Ultimate monolingual data

Here PPL is the perplexity of a language model for sentence, λ is an empirical value between 0.2–0.8 depending on the language pair, such as the source language is English, and the target language is Croatian, then our empirical value of λ is 0.7. Finally, we consult Parallel Corpus Filtering Zhang et al. (2020) for finetuning a multilingual high-resource corpus classifier using mBERT (Gonen et al., 2020) to get our ultimate training data described in Table 1.

2.1.2 Monolingual Data

As we need a multilingual language model to filter low-quality corpus and create synthetic parallel text, we collect all high-quality monolingual corpus from News-Commentary, europarl, and news-crawl for all languages if available. For medium and low-quality resources, we use all available monolingual data from the shared task, and filter according to the above steps (where applicable). The ultimate monolingual data is described in Table 2.

Module	Big	Large
Layers	12	24
Attention Heads	16	16
Embedding Size	1024	1024
FFN Size	2048	4096
Shared Vocab	True	True

Table 3: Hyper-parameters and model sizes of different models used in our systems.

2.2 Tokenizer

We use sentencepiece (Kudo and Richardson, 2018) to train a multilingual subword tokenizer. To represent the low-resource languages better, we follow Tran et al. (2021)'s settings, sampling text with temperature 5. Especially, for Yakut, we take monolingual data into account, since it's an extremely low-resource. Finally, For bilingual models, we used a vocabulary size of 32,000, and for multilingual models, we used 100,000.

We also apply subword regularization methods (Provilkov et al., 2019; Raffel et al., 2020) when tokenizing text. For low- and medium-resource directions, we apply BPE dropout on both the source and target sides and double the corpus size. And for high-resource directions, we only use it on the source side and don't do data augmentation stuff.

2.3 Model Architectures

Similar to Tran et al. (2021), we train two separate models: Many to English, or one system encompassing every language translated into English, and Many to Many directions, or one for English into every language and other non-English directions. Due to the very late release of the Yakut to the Russian corpus, we apply simple finetuning and backtranslation in this direction. For Chinese to and from English, we train a separate model. Because we are native speakers of Chinese and good at English, we introduce about 20 million high-quality private corpus [2]

Dense Multilingual Model Our model settings are empirically designed based on Transformer (Vaswani et al., 2017). We introduce two model architectures seen in Table 3. All models are implemented on top of the open-source toolkit Fairseq (Ott et al., 2019).

We also train three bilingual models: English to/from Chinese, and French to Germany. The aim is to compare how similarities among different languages will influence multilingual model. Due to the limitation of computing resources, we do not test in other language directions.

Language Specific Adapter In brief, an adapter layer is a dense layer with residual connection and non-linear projection. The hyperparameter b is the dim size of the inner dense layers. With a large set of globally shared parameters and small interspersed task-specific layers, adapters allow us to train and adapt a single model for a huge number of languages. Bapna et al. (2019) shows translation performance improvement in multilingual models with residual adapters. So after training the dense multilingual model, we add adapters for each language direction and apply further training and finetuning on these adapter layers. In detail, for high-resource directions, we add a larger adapter (b=4096). As for medium-resource, we set b=2048 and for the low-resource, we set b=1024.

2.4 Optimization Tricks

Backtranslation As shown in previous news task submissions, such as Tran et al. (2021) and Wang et al. (2021), backtranslation can significantly improve the BLEU score in low- and medium-resource language directions. We find no significant improvement in high-resource directions. And for some "X-en" high-resource directions, like zh-en shows in Table 4, backtranslation even lower the BLEU score. For this reason, we collect monolingual data for low- and medium-resource directions. All backtranslation data are generated by our well-trained multilingual model with Transformer Big settings. We use this generated data to train models with "Large" settings.

Finetune We use in-domain finetuning to further improve the model performance, which has proven effective on previous news translation tasks. We construct different types of finetuning data with the following approaches. Li et al. (2020); Wang et al. (2021) shows that low-frequency words frequency words are mostly domain-specific nouns, etc., which may indicate the topic directly. On the other hand, this year the shared task has changed from a news domain to a general translation task. We think finetuning our model by previous in-domain news data may be harmful to our model.

[2] We have a data group and a translation review team. First, we collect public monolingual data to make it multilingual. Second, we have a cooperative corpus or terminology base with our clients. With the consent of our clients, some non-public corpora and terminology are used for training.

	Test Set	Big Model	Large Model	+BT	+Adapter	+Finetune	+LM Rerank
cs-en	wmt21	23.0	23.9	—	24.2	24.8	25.2
uk-en	flore101	35.7	35.9	36.1	37.0	37.0	37.5
ja-en	wmt21	21.5	21.8	24.0	27.2	28.0	28.0
de-en	wmt21	29.4	29.9	—	30.0	32.1	32.3
ru-en	wmt21	30.1	31.3	32.5	34.0	37.5	37.9
en-cs	wmt21	15.7	17.0	—	20.4	20.2	21.3
en-uk	flore101	24.1	24.5	27.1	28.0	28.9	29.0
en-ja	wmt21	16.9	18.0	22.5	22.8	25.0	25.1
en-de	wmt21	24.4	24.8	25.0	25.0	27.1	27.3
en-ru	wmt21	20.6	21.0	21.1	23.4	24.2	25.6
en-hr	flore101	25.8	26.4	28.9	29.3	30.0	30.3
cs-uk	flore101	19.8	21.3	25.0	25.9	26.2	26.6
uk-cs	flore101	20.8	22.0	24.1	24.3	24.3	24.5
fr-de	wmt21	35.8	36.1	37.3	39.1	39.0	39.1
zh-en	wmt21	31.4	32.0	31.7	—	34.0	34.1
en-zh	wmt21	33.0	33.4	35.1	—	35.5	35.7
Avg Incremental	—	—	0.70	2.99	2.40	4.11	4.47

Table 4: Evaluation result on dev dataset. The inside of the dividing line represents the same model. We train X-en, en-X X-X, zh-en, and en-zh models separately. All translations are generated by beam search with beam size 5. All the models are the average of the final 5 checkpoints.

So we follow Li et al. (2020); Wang et al. (2021)'s strategies to select topic-related data based on a testset. We use the selected data for further finetuning. We experimented with the 2022 news development set and apply it directly to the 2022 test set.

Language Model Reranking Following Yee et al. (2019); Tran et al. (2021), we train language models and apply noisy channel reranking to the outputs of our final system. Unlike Tran et al. (2021), which trains a separate language model for each language, we train a multilingual language model for all languages to evaluate whether the multilingual language models can also improve the quality of translations.

Model Ensemble Model ensemble is a widely used technique in previous WMT shared tasks. To deal with biases toward recent training data, it is common to average parameters across multiple checkpoints of a model. We always average the last 5 checkpoints during training. During finetuning, we tune this hyperparameters (num epoch and num average checkpoints) on the development set and use it directly on the test set of wmt22.

3 Experiment

We conduct experiments to quantify the impact of each component in our system. The evaluation conduct on newstest2021 or development set on wmt22 using SacreBLEU (Post, 2018).

3.1 Settings

Every single model is trained on 8 NVIDIA A100 GPUs, each of which has 40 GB of memory. We also employ large batching with larger learning rates (Ott et al., 2018). We set the max learning rate to 0.0005 and warmup steps to 10000. All the dropout probabilities are set to 0.1. To speed up the training process, we conduct training with a half-precision floating point (FP16). During training multilingual, we add both source-side language tags and target-side language tags to leverage the gap between different language pairs. Following Tran et al. (2021), we divide data into multiple shards and downsample data from both high-resource directions and synthetic backtranslated with each training epoch using one shard.

3.2 Multilingual Models Result

We mainly evaluate our model and method on the wmt21 test set and flore101 dataset (Goyal et al., 2021). We analyze each aspect in our final submission and the cumulative effect. The effect of each component is shown in Table 4.

According to our experimental results, increasing the model capacity, increasing the sparsity of the model (adding a specific set of Adapter Layer

	en-zh	zh-en	fr-de
Bilingual Model Big	33.0	31.4	32.6
Multilingual Model Big	31.9	30.2	35.8

Table 5: BLEU score on wmt21 test set. All result is based on Big Model without any optimization

for each speech direction), fine-tuning the training set by extracting more relevant corpus based on the original text of the test set, and using the language model for reranking is effective on all language directions and the test set. Backtranslation is particularly effective in low- and medium-resource languages. Although the improvement of BLEU value by backtranslation on high-resource languages is not obvious or even worse, the average improvement by backtranslation is as obvious in a comprehensive view, with an average improvement of 1.68 BLEU per language direction.

Because this year's task is a general-purpose machine translation, rather than the news domain machine translation task of previous years, we are not submitting translation results that validate the optimal model on the development set, but rather the results of model fine-tuning on selected domain data after the release of the test set.

3.3 Distant Language Pairs Analysis

As shown in Tran et al. (2021), the multilingual model can significantly improve the BLEU score of medium- and low-resource language directions. For high-resource language directions, there are no significant enhancements. For high-resource languages, such as en-de, the BLEU score decreases slightly, and this is even more severe for distant language directions, en-ja, and en-zh for example. To compare the influence of the distance of the language family on the multilingual model. We train bilingual models for en-zh, zh-en, and fr-de. The test result is shown in Table 5. Since most of the language directions of wmt22 are Indo-European, distant languages, Chinese and Japanese for example, cannot benefit from the knowledge transfer additive of other languages, while the parameter capacity of the multilingual model is limited. These factors lead to poor results. Overall, when training multilingual models, languages with similar language families should be trained together, instead of putting all the languages together.

Task	**BLEU**	**Task**	**BLEU**
`cs-en`	25.3⋆	`en-cs`	26.3
`de-en`	33.4	`en-de`	36.1
`ja-en`	22.8	`en-ja`	39.4
`ru-en`	45.2*	`en-ru`	32.6
`uk-en`	44.6⋆	`en-uk`	29.5
`zh-en`	28.1	`en-zh`	48.3
`fr-de`	41.8	`en-hr`	18.2⋆
`uk-cs`	36.5	`cs-uk`	38.3
`ru-sah`	15.3⋆	`sah-ru`	7.1⋆

Table 6: Our final submission results in 18 tasks. ⋆ represents the best score in the automatic evaluation. Note that the result is based on the submission website https://ocelot-wmt22.mteval.org/, not the official final result.

3.4 Submission Results

The results we finally submitted are shown in Table 6. We participate in 18 tasks this year. On the whole, all of our systems performed competitively, especially for Many-to-English directions. Yakut to/from Russian tasks is added bonus. Few teams participate in these two tasks.

4 Conclusion

In this paper, we described Lan-Bridge's submission to the WMT2022 General Translation shared task. Our main exploration was using a multilingual model to train different language pairs. It shows that the multilingual model can achieve state of art results in both high- and low-resource language directions. Meanwhile, we found that the multilingual model worked better for languages from the same or close language families than languages from distant language families. Finally, for extremely low-resource languages, even a multilingual model can boost their performance of them, but the translation is still far from usable.

5 Acknowledgments

We would like to thank our translator colleagues (Ruicheng Liu, and Qingtong Yang) for their assistance with linguistic knowledge and partners development teams (Jun Zhang et al.) for their foundational contributions to this project. We would also like to thank Yafu Li, and Yulong Chen from Westlake University for helpful discussions early on in the project.

References

Ankur Bapna, Naveen Arivazhagan, and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. *arXiv preprint arXiv:1909.08478*.

Chao Bei, Hao Zong, Conghu Yuan, Qingming Liu, and Baoyong Fan. 2019. Gtcom neural machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 116–121.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*.

Hila Gonen, Shauli Ravfogel, Yanai Elazar, and Yoav Goldberg. 2020. It's not greek to mbert: Inducing word-level translations from multilingual bert. *arXiv preprint arXiv:2010.08275*.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Zuchao Li, Hai Zhao, Rui Wang, Kehai Chen, Masao Utiyama, and Eiichiro Sumita. 2020. Sjtu-nict's supervised and unsupervised neural machine translation systems for the wmt20 news translation task. *arXiv preprint arXiv:2010.05122*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation (WMT)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alan Aspuru-Guzik, and Alex Zhavoronkov. 2020. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Frontiers in Pharmacology*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2019. Bpe-dropout: Simple and effective subword regularization. *arXiv preprint arXiv:1910.13267*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin. 2019. Ccmatrix: Mining billions of high-quality parallel sentences on the web. *arXiv preprint arXiv:1911.04944*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. Facebook ai wmt21 news translation task submission. *arXiv preprint arXiv:2108.03265*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Longyue Wang, Mu Li, Fangxu Liu, Shuming Shi, Zhaopeng Tu, Xing Wang, Shuangzhi Wu, Jiali Zeng, and Wen Zhang. 2021. Tencent translation system for the wmt21 news translation task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 216–224.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Conference on Empirical Methods in Natural Language Processing*.

Boliang Zhang, Ajay Nagesh, and Kevin Knight. 2020. Parallel corpus filtering via pre-trained language models. *arXiv preprint arXiv:2005.06166*.

Manifold's English-Chinese System at WMT22 General MT Task

Chang Jin
Soochow University*
cjin@stu.suda.edu.cn

Tingxun Shi
OPPO
shitingxun@oppo.com

Zhengshan Xue
OPPO
xuezhengshan@oppo.com

Xiaodong Lin
Rutgers University
lin@business.rutgers.edu

Abstract

Manifold's English-Chinese System at WMT22 is an ensemble of 4 models trained by different configurations with scheduled sampling-based fine-tuning. The four configurations are DeepBig (XenC), DeepLarger (XenC), DeepBig-TalkingHeads (XenC) and DeepBig (LaBSE). Concretely, DeepBig extends Transformer-Big to 24 encoder layers. DeepLarger has 20 encoder layers and its feed-forward network (FFN) dimension is 8192. TalkingHeads applies the talking-heads trick. For XenC configs, we selected monolingual and parallel data that is similar to the past newstest datasets using XenC, and for LaBSE, we cleaned the officially provided parallel data using LaBSE pretrained model. According to the officially released autonomic metrics leaderboard[1], our final constrained system ranked 1st among all others when evaluated by bleu-all, chrf-all and COMET-B, 2nd by COMET-A.

1 Introduction

This report describes Manifold's machine translation system submitted to WMT 22 English→Chinese general domain translation task. The general domain translation task of WMT is a new task set up this year, replaces the time-honored news translation task. However, as the newstest datasets released previously were created by professional translators manually, they were considered to have better quality compared with the web crawled dataset. Therefore, generally our strategy is selecting data from the provided datasets according to their similarity between past WMT test sets, and training big models with different architectures or training methods based on the selected data.

2 Data Preprocessing

For officially released bilingual data, we merged ParaCrawl v9, News Commentary v16, Wiki Titles v3, UN Parallel Corpus v1.0, CCMT corpus and WikiMatrix corpus together, then applied multiple rules to preprocess and filter the officially released bilingual data, including

- Normalizing punctuation.
- Removing unprintable characters.
- Tokenizing texts. jieba[2] was applied for Chinese texts, and moses tokenizer[3] was applied for English texts.
- Removing all the sentence pairs whose source text or target text tokens count exceeds 150, and the segment pairs with a source-target token ratio lower than 2/3 or higher than 3/2.
- Removing all the sentence pairs that belong to other languages. We applied the fasttext model (Joulin et al., 2016)[4] as our language identifier.
- Deduplicating the dataset.

3 Basic models

After applying the aforementioned preprocessing and filtering rules, 30M segment pairs were kept [5]. For the kept data, we adopted two different selection strategies:

1. Following NiuTrans' system submitted to WMT21 (Zhou et al., 2021), we selected 12M segment pairs from the kept data which are

[*] Work done during Jin's internship in OPPO.

[1] https://github.com/wmt-conference/wmt22-news-systems/blob/main/scores/automatic-scores.tsv

[2] https://github.com/fxsjy/jieba

[3] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl

[4] https://dl.fbaipublicfiles.com/fasttext/supervised-models/lid.176.bin

[5] We did not make use of official back-translated corpus

Config Item	Big	Deep	Deeper	DeepBig	DeepLarger
# Encoder Layer	6	30	40	24	20
# Attention Heads	16	8	8	16	16
Embedding Size	1024	512	512	1024	1024
FFN Size	4096	2048	2048	4096	8192
Pre-Norm	No	Yes	Yes	No	No

Table 1: Main configurations for different architectures we applied for basic models. Decoder layers numbers were fixed to 6 for all the architectures if not specially mentioned.

most similar to our validation set (i.e. newstest2020enzh data) using XenC (Rousseau, 2013). This part of data will hereinafter be referred as "XenC" for short.

2. We further cleaned the 30M segment pairs using LaBSE (Feng et al., 2022)[6], set the threshold to 0.7 according to our experience in filtering out un-aligned data. After this step, 24.3M segment pairs were kept. This part of data will hereinafter be referred as "LaBSE" for short.

Config Item	Value
dropout	0.3
learning rate	0.0005
max tokens	4096
warmup init lr	1e-7
warmup steps	4000
label smoothing	0.1
num max updates	300,000
update frequency	8

Table 2: Hyper-parameters for training models. All experiments were conducted on 4 or 8 Tesla V100 GPUs.

We applied BPE-subword (Sennrich et al., 2016)[7] to divide tokens into subwords. BPE codes were learned jointly with 32K merge operations but dictionaries for the source language and target language are generated separately.

Basic models were trained to generate synthetic, pseudo bilingual segment pairs for the next step. As is discovered by Zeng et al. (2021), sub-model diversity is a key factor to enhance the performance of ensemble model. Therefore, we trained various Transformer models (Vaswani et al., 2017) applying different architectures. For very deep Transformers, we followed the suggestion given by DLCL (Wang et al., 2019) to use Pre-Norm. Main configurations for the architectures we trained are listed in Table 1. Other hyper-parameters for training models are listed in Table 2 (same for all the experiments we took in the contest).

For each architecture X listed in Table 1, we also trained its talking-heads attention variant (Shazeer et al., 2020) to further increase model diversity. Such variants will be denoted as "X-th" in the following part of the report. All the models were developed and trained using fairseq (Ott et al., 2019) [8]

4 Data Augmentation

Previous studies show that adding synthetic data can help to boost the performance of machine translation systems (Edunov et al., 2018) (Hoang et al., 2018). We adopted four data augmentation methods during the contest, including back-translation (with sampling), forward-translation, sequence knowledge distillation and R2L translation. For R2L translation, we reversed the token sequences of inputs, e.g. converted "This is a book ." to ". book a is This", and left the target sentences unchanged.

We selected 12M sentences from officially released English and Chinese monolingual datasets respectively[9], also applying XenC algorithm on them. The selected monolingual data was preprocessed by the similar pipeline presented in section 2, with the difference on skipping the steps to filter unaligned bilingual data. The reason behind selecting 12M sentences is we expect that the data from each monolingual dataset has the same size compared with the bilingual training data, as Edunov et al. (2018) indicated.

The preprocessed Chinese monolingual data was

[6] We downloaded the pretrained model from transformers official website on October 11th., 2021. As the model was released before February 2022, the constrained system requirement is not violated.

[7] https://github.com/rsennrich/subword-nmt

[8] https://github.com/facebookresearch/fairseq

[9] For English, we combined News Crawl, News Discussions, News Commentary and Europarl v10 corpus together; for Chinese, we combined News Crawl, News Commentary and Common Crawl corpus.

No.	Method	Data Size	Newstest 2020	Newstest 2021
1	Deep baseline model	12M	42.7	32.4
2	1 + Forward-translation (ForT)	24M	44.7 (+2.0)	33.3 (+0.9)
3	1 + Top-p back-translation (topp BT)	24M	45.8 (+3.1)	32.9 (+0.5)
4	1 + R2L KD + R2L ForT	36M	44.2 (+1.5)	33.5 (+1.1)
5	Sequence KD	12M	42.7 (-)	32.5 (+0.1)
6	5 + ForT + topp BT	36M	45.3 (+2.6)	33.7 (+1.3)
7	4 - 1 + 6	60M	45.5 (+2.8)	33.8 (+1.4)

Table 3: Model performances when applying different methods. All the models are trained by Deep architecture depicted in Table 1. Our validation dataset is from newstest2020 (Barrault et al., 2020) and test dataset is from newstest2021 (Akhbardeh et al., 2021).

No.	Method	Newstest 2020	Newstest 2021
1	Fine-tuned Deep	46.2	34.6
2	DeepBig	47 (+0.8)	35 (+0.4)
3	DeepLarger	47.5 (+1.3)	35.7 (+1.1)
4	DeepBig-th	47.6 (+1.4)	35.4 (+0.8)
5	DeepBig (LaBSE)	47.6 (+1.4)	35.5 (+0.9)
6	Ensemble of 2, 3, 4 and 5	48.1 (+1.9)	36.2 (+1.6)

Table 4: Detailed performance information of the four bigger model and the final ensemble model. All the improvements are based on the baseline model (Deep model shown in the last line of Table 3, fine-tuned using past newsdev/test datasets, by applying decoder steps based schedule sampling as regularization). All sub-models (No. 2 to 5) are also fine-tuned by the same datasets applying the same regularization method.

then back-translated into English by an ensemble model, which is composed by a Big model, a Deep model and a Big-th model, all trained with the bilingual XenC Chinese→English corpus. Inspired by some ideas of Burchell et al. (2022), we performed top-p (nucleus) sampling (Holtzman et al., 2019) in the process of back-translation to import some noises, and set topp to 0.9. In this way, 12M Pseudo English-Chinese segment pairs were constructed.

The preprocessed English monolingual data was forward-translated into Chinese, leading to another 12M English-Pseudo Chinese dataset. The ensemble model used to generate data has the same architecture with the back-translation model introduced above, the only difference is it is trained by the parallel XenC English→Chinese corpus.

We further applied sequence-level knowledge distillation (Kim and Rush, 2016) to distill this ensemble model by translating English sentences of the parallel XenC corpus into pseudo Chinese. Furthermore, we also trained a Deep model with the reversed parallel XenC data to get a right-to-left (R2L) model, and generated forward-translation and the results of knowledge distillation using reversed monolingual and parallel English corpora. After having acquired these different synthetic datasets, we trained Deep models by various combinations of them. The concrete model performance is shown in Table 3.

5 Fine-tuning and Bigger Models

After having acquired extra data by back/forward-translation, knowledge distillation, and R2L data augmentation, we experimented on several other methods to further improve our system.

Fine-tuning. We fine-tuned our Deep model using the combination of newsdev2017, newstest2017, newstest2018, and newstest2019 datasets. As the dataset used for fine-tuning is quite small, we applied decoder steps based scheduled sampling (Liu et al., 2021) as a means of regularization. We set k to 0.99. With this step, a 0.7 BLEU gain has been brought on the validation set and 0.8 BLEU gain on the test set.

Ensemble of bigger models. We trained three bigger models on the 60M XenC Dataset (configuration No. 7 in Table 3), including a DeepBig model, a DeepLarger model and a DeepBig-th model. Furthermore, we replaced the distilled data in the XenC Dataset (12M) with the LaBSE data (24M), and got a dataset containing 72M segment pairs which is used to train another DeepBig model. We fine-tuned these four models using past newsdev/test datasets and applied decoder steps based schedule sampling, then made an ensemble model comprised of them. Table 4 lists the detailed performance of the bigger models and their ensemble model.

Post-processing. We converted punctuation from half-width symbols to full-width symbols for the generated results.

6 Conclusion

In this report, we describe our Manifold English→Chinese system submitted to WMT 22 general translation task. The core idea of our system is to train various big Transformer models utilizing in-domain (actually news domain) data, based on which an ensemble model is created. Although most training data belongs to a special domain, we still achieved compelling results in the final submission, i.e. our final system ranked first among the constrained systems, evaluated by BLEU score based on the two references.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussà, Cristina España Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (wmt21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (wmt20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Laurie Burchell, Alexandra Birch, and Kenneth Heafield. 2022. Exploring diversity in back translation for low-resource machine translation. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pages 67–79, Hybrid. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd workshop on neural machine translation and generation*, pages 18–24.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.

Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. 2021. Scheduled sampling based on decoding steps for neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3285–3296.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Anthony Rousseau. 2013. Xenc: An open-source tool for data selection in natural language processing. *The Prague Bulletin of Mathematical Linguistics*, 100(1):73.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Noam Shazeer, Zhenzhong Lan, Youlong Cheng, Nan Ding, and Le Hou. 2020. Talking-heads attention. *arXiv preprint arXiv:2003.02436*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822.

Xianfeng Zeng, Yijin Liu, Ernan Li, Qiu Ran, Fandong Meng, Peng Li, Jinan Xu, and Jie Zhou. 2021. Wechat neural machine translation systems for wmt21. In *Proceedings of the Sixth Conference on Machine Translation*, pages 243–254.

Shuhan Zhou, Tao Zhou, Binghao Wei, Yingfeng Luo, Yongyu Mu, Zefan Zhou, Chenglong Wang, Xuanjun Zhou, Chuanhao Lv, Yi Jing, et al. 2021. The niutrans machine translation systems for wmt21. In *Proceedings of the Sixth Conference on Machine Translation*, pages 265–272.

CUNI Submission in WMT22 General Task

Josef Jon
Charles University
surname@mail.ufal.mff.cuni.cz

Abstract

We present CUNI-Bergamot submission for WMT22 General translation task. We compete in English → Czech direction. Our submission further explores block backtranslation techniques. In addition to the previous work, we measure performance in terms of COMET score and named entities translation accuracy. We evaluate performance of MBR decoding compared to traditional mixed backtranslation training and we show possible synergy when using both of the techniques simultaneously. The results show that both approaches are effective means of improving translation quality and they yield even better results when combined.

1 Introduction

This work focuses on exploring of two methods used in NMT in order to improve translation quality: backtranslation and Minimum Bayes Risk decoding using neural-based evaluation metric as a utility function. The methods used and related work are presented in the following section. In next section we describe our experimental setting and results.

2 Methods

We describe methods we used to build our system in this section.

2.1 Block backtranslation

The translation quality of NMT depends heavily on the amount of parallel training data. It has been shown that the authentic bilingual data can be partially supplemented by synthetically parallel, machine translated monolingual text (Bojar and Tamchyna, 2011; Sennrich et al., 2016; Xie et al., 2018; Edunov et al., 2018). Often the synthetic and authentic parallel data are mixed in the training dataset, but previous research shows that simply mixing the two types of text does not yield optimal translation quality. We are using block backtranslation (*block-BT*) in similar configuration to Popel et al. (2020). This method creates blocks of parallel and synthetic data and presents them to the neural network separately, switching between the two types during the training. Since in last year's WMT, the submission using block-BT by Gebauer et al. (2021) did not find any improvements, presumably due to improperly chosen block size, we decided to verify effectiveness of this method once again.

Averaging type Previous work on *block-BT* shows the importance of averaging the checkpoints to combine information from different blocks of training data in order to obtain good performance. We compare checkpoint averaging with another method of combining older sets of model's parameters with the current one – *exponential smoothing*. After each update u, the current parameters Θ_u are averaged (with smoothing factor α) with parameters after the previous update Θ_{u-1}:

$$\Theta_u = \alpha\Theta_u + (1-\alpha)\Theta_{u-1}$$

Previous work by Popel (2018) contains experiments with exponential averaging, but only on the level of already saved checkpoints, not online during the training after each update as for our work.

Minimum Bayes Risk Decoding NMT models predict conditional probability distribution over translation hypotheses given a source sentence. To select the most probable translation under the model (mode of the model's distribution), an approximation of MAP (*maximum-a-posteriori*) decoding is used, most commonly the beam search (Graves, 2012). However, beam search and MAP decoding in general has many shortcomings described in recent work (Stahlberg and Byrne, 2019; Meister et al., 2020) and other approaches have

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 280 - 289
December 7-8, 2022 ©2022 Association for Computational Linguistics

been proposed to generate a high-quality hypothesis from the model.

One of them, MBR (Minimum Bayes Risk) decoding (Goel and Byrne, 2000; Kumar and Byrne, 2004), has been proposed as an alternative to MAP. MBR does not produce a translation with the highest probability, rather a translation with the best value of utility function. This utility function is usually an automatic machine translation evaluation metric. However, to optimize towards best utility function value, it would necessary to know the ideal selection of hypothesis. In case of MT, that would mean a perfect, best possible translation, which of course is not known during the translation process. For this reason, an approximation of the ideal translation is used, based on the model's probability distribution (Bryan and Wilker, 2021). This can be implemented as generating a list of hypotheses (e.g. using sampling or beam search) and then computing utility function of each hypothesis using all the other hypotheses as the ideal translation approximation (i.e. as references). This approximation of MBR decoding can be seen as consensus decoding – the hypothesis that is the most similar to all the others is chosen.

Even though MBR is able to optimize towards many metrics and increase the scores, these gains did not translate into better human evaluation of the final translations, when using traditional metrics based on surface similarities like BLEU. Recent successes in development of novel metrics for machine translation has renewed interest in this method. (Amrhein and Sennrich, 2022a; Freitag et al., 2021; Müller and Sennrich, 2021).

3 Experiments

In this section we present our experimental setup and results.

3.1 Tools

We tokenize the text into subwords using FactoredSegmenter[1] and SentencePiece (Kudo and Richardson, 2018). We use MarianNMT (Junczys-Dowmunt et al., 2018) to train the models. BLEU scores are computed using SacreBLEU (Post, 2018), for COMET scores (Rei et al., 2020) we use the original implementation[2].

3.2 Datasets

We train English-Czech NMT models for our experiments. We train our models on CzEng 2.0 (Kocmi et al., 2020). We use all 3 subsets of CzEng corpus: the originally parallel part, which we call *auth*, Czech monolingual data translated into English using MT (*csmono*) and English monolingual data translated into Czech using MT (*enmono*). We use `newstest2020` (Barrault et al., 2020) as our dev set and `newstest2021` (Akhbardeh et al., 2021) as our test set.

For experiments concerning translation of named entities, we used a test set originally designed for Czech NLG in restaurant industry domain[3](Dušek and Jurčíček, 2019). It contains sentences which include names of restaurants and addresses in Czech and their translations in English. We will call this test set the `restaurant` test set.

3.3 Models

We train Transformer-base (which we denote *base*) and Transformer-big (*big 6-6*) models with standard parameters (Vaswani et al., 2017) as preconfigured in MarianNMT. For the largest model (*big 12-6*), we use Transformer-big with 12 encoder layers and depth scaled initialization (Junczys-Dowmunt, 2019; Zhang et al., 2019)[4]. We also used learning rate of $1e{-}4$ for the 12 layer model instead of $3e{-}4$, which was used for other models. We trained all models for at least 1.4M updates. After that, we computed validation BLEU scores every 5k updates and we stopped if the score did not improve for 30 consecutive validations. We trained the models on heterogenous grid server, which includes combinations of Quadro RTX 5000, GeForce GTX 1080 Ti, RTX A4000 and GeForce RTX 3090 cards. Typical training time on 4 108Ti of the base models for 1.4M updates was 7 days.

3.4 Block-BT settings

For all our experiments, we create a checkpoint each 5k updates and we vary only the size of the blocks during which the training data have the same type (20k, 40k, 80k and 160k updates). The size is the same for all block types. We circle through the block types in the following order: *auth*→*csmono*→*auth*→*enmono*.

[1] `https://github.com/microsoft/factored-segmenter`

[2] `https://github.com/Unbabel/COMET`

[3] `https://github.com/UFAL-DSG/cs_restaurant_dataset`

[4] Training scripts available at: `https://github.com/cepin19/wmt22_general`

For checkpoint averaging, we average 8 checkpoints. For exponential smoothing, we use default Marian configuration ($\alpha = 0.001$, but there are some slight modifications based on number of updates since start of the training and batch size).

We also look at the effects of using only back-translation, or both back- and forward-translation.

3.5 Block-BT results

Training regime and averaging method First, we compare different training regimes: *mixed-BT*, where all the training datasets are concatenated and shuffled together and *block-BT* with 40k updates long blocks and two possible averaging types – exponential smoothing (*exp*) or checkpoint averaging (*avg8*).

Figure 1 shows behavior of BLEU and COMET scores on `newstest2020` during the training for these configurations. We opt to present the interval between 480k and 1280k updates. We chose the lower bound because the behavior is more stabilized than in the beginning of the training and the upper bound because all the models were trained for at least 1400k updates and 1280k is the nearest lower multiplicative for the largest block size. *40k block* curve represents a model without any averaging, *40k block avg8* is a model trained without exponential smoothing, but each checkpoint was averaged with 7 previous checkpoints for the evaluation, *40k block exp* model was trained with continuous exponential smoothing. Finally, we also experimented with combination of both - trained with exponential smoothing and averaged after the training. The combination does not improve over the separate averaging techniques and we omitted the curve from the figure to make it more readable.

In both metrics, *block-BT* with either form of averaging outperforms *mixed-BT* training. Without any averaging, the advantage of *block-BT* over *mixed-BT* is smaller. Type of averaging does not seem to play a large role – checkpoint averaging, exponential smoothing and their combination yield very similar best scores. The best scores on `newstest2020` for each combination of parameters are presented in Table 1.

The curves for checkpoint averaging and exponential smoothing behave similarly, with exponential averaging reacting faster to change of the block. Additionally, the *avg8* models have higher peaks in *enmono* (red) blocks, especially for BLEU scores. The shape of the curves could be tuned by changing frequency of saving checkpoints and number of checkpoints to be averaged for checkpoint averaging method, or by changing the α factor for exponential smoothing.

There are differences in behaviour between BLEU and COMET score curves. Most notably, COMET is less sensitive to transition from *auth* (green) to *csmono* (blue) blocks. We hypothesize this is caused by lower sensitivity of COMET score to wrong translation of named entities and rare words (Amrhein and Sennrich, 2022a). We present further experiments in this direction later.

Block size We asses influence of block size for both of the two averaging methods. We compare block sizes of 20k, 40k, 80k and 160k updates. Behaviour of COMET and BLEU scores is presented in Figures 2 and 3 for exponential smoothing and checkpoint averaging, respectively. The best scores are again shown in Table 1.

We see that 20k block size yields noticeably worse results when using checkpoint averaging that the other sizes. The negative effect of the small block size is less pronounced when using exponential smoothing, yet still present. Other block sizes perform similarly in both metrics. This results is expected, since for 8-checkpoint averaging with 5k updates checkpointing interval, it is necessary to have a block size of at least 40k updates to fit all the 8 checkpoints and thus explore all possible ratios of *auth* and *mono* data.

Reverse direction For the reverse direction, Czech to English, we performed less extensive evaluation. We only compare *mixed*, *block-BT* with 40k blocks and either exponential smoothing or checkpoint averaging. Behavior of the metrics is shown in Figure 4 and final best scores on `newstest2020` are presented in Table 2. *Block-BT* still outperforms *mixed* training, but by a smaller margin than in the other direction.

Backtranslation direction We also evaluate influence of using only backtranslations as additional synthetic data (monolingual data in target language to automatically translated to source language) or adding also forward translations (from source language to target target) and we present the results in Table 3. Interestingly the results show large gains in both BLEU and COMET when using forward translation. We hypothesize this is caused by the good quality of the model used to perform the forward translation. In such case, the translation

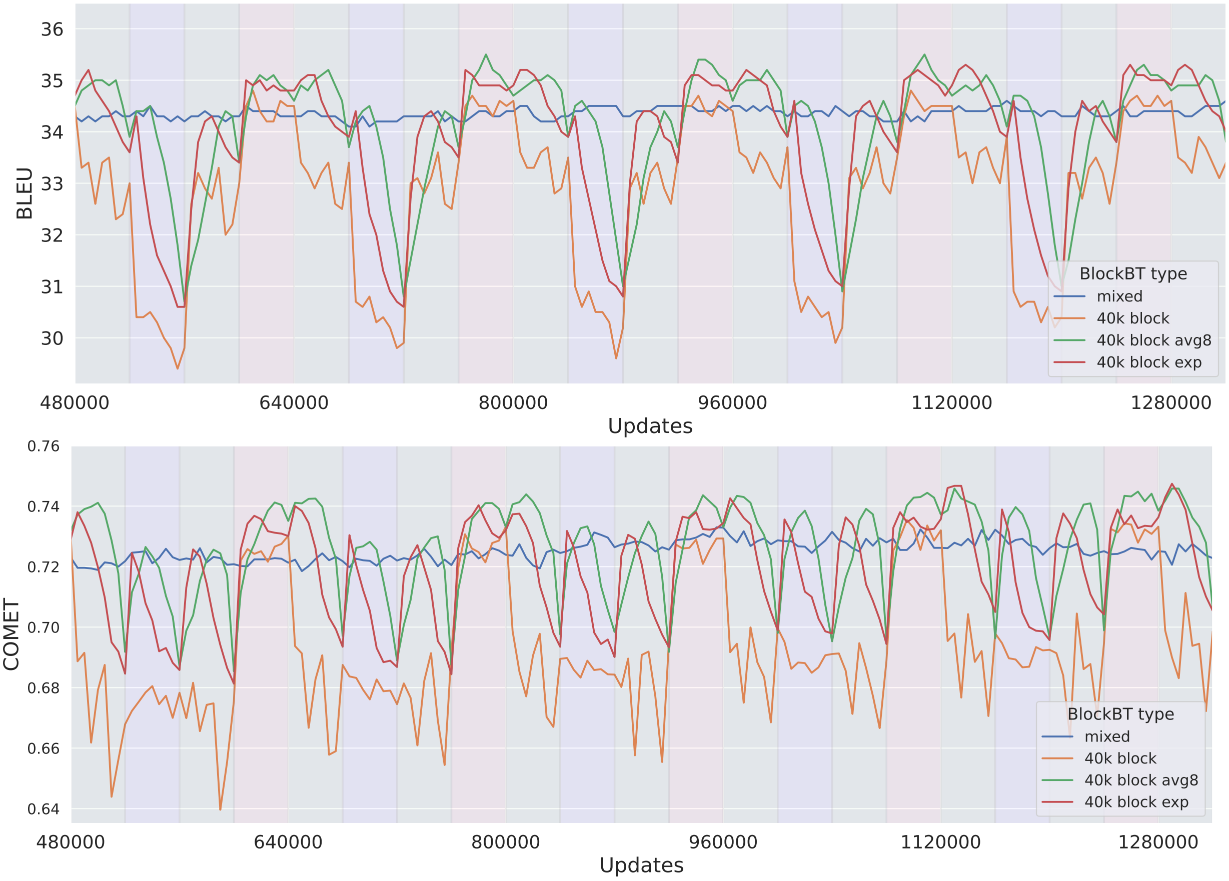


Figure 1: Comparison of different training regimes for EN-CS translation on newstest20 in terms of BLEU (top) and COMET (bottom). Background colors for block-BT regime show which part of training data was used for given part of the training. Green means authentic parallel data, blue is CS->EN backtranslation and red is EN->CS forward translation.

model assumes the role of the teacher in teacher->student training and might lead to a good quality results.

Named entities test sets From anecdotal evidence, we have seen that checkpoints with large influence of backtranslated data perform worse on named entities translation and COMET and BLEU scores might not reflect this drop of accuracy. We evaluate the models in terms of accuraccy of named entitiy translation on the `restaurant` test set. We selected Czech to English direction, since the evaluation is easier given lower morphological richness of target language. Figure 5 shows comparison of behavior of named entities translation accuracy on the restaurant test set and COMET and BLEU scores on `newstest2020` for exponential smoothing and checkpoint averaging. NE accuracy peaks towards the end of *auth* regions (green). Both COMET and BLEU scores peak also during the *auth* part of the training, but, especially for COMET, the peak occurs in earlier stages after the switch to *auth*. Overall, BLEU curve correlates better with the NE accuracy curve. We hypothesize this might be related to the fact that COMET was found to be insensitive to named entities errors by Amrhein and Sennrich (2022b).

However, it seems that the shift between the accuracy and the other two metrics is not too large in our settings and choosing the best performing model in terms of either COMET or BLEU should not hurt NE translation by a large amount. We further investigate that in Table 4 – we chose the checkpoint with best COMET (first row) and best BLEU (second row) on the `newstest2020` and the checkpoint with best NE translation accuracy on the restaurant test set (third row). We compute all three metrics for these three models. The best COMET checkpoint obtains accuracy of 60.7% on the restaurant test set, the best BLEU checkpoint reaches accuracy of 62.9%, while the best accuracy reached by any checkpoint is 63.6%.

Model size	Block size	Avg type	update (k)	BLEU	update (k)	COMET
base	mixed	exp	1340	34.7	1760	0.7337
	mixed	exp+avg8	1365	34.7	965	0.7326
	20k	-	1360	34.6	640	0.7324
		exp	410	34.9	725	0.7406
		avg8	660	34.8	1385	0.7349
		exp+avg8	420	34.9	735	0.7399
	40k	-	610	34.8	1415	0.7363
		exp	1130	35.3	1290	**0.7474**
		avg8	780	35.5	1420	0.7462
		exp+avg8	1150	35.5	1075	0.7466
	80k	-	1250	34.9	960	0.7393
		exp	1210	35.2	1450	0.7447
		avg8	985	35.5	665	**0.7474**
		exp+avg8	585	35.3	1150	0.7455
	160k	-	1130	34.9	1210	0.7387
		exp	1125	35.3	1285	0.7453
		avg8	1135	35.5	1305	0.7467
		exp+avg8	1145	35.3	1310	**0.7473**
big 6-6	40k	exp	445	35.4	1125	0.7546
		exp+avg8	300	35.4	1310	0.7567
big 12-6	40k	exp	130	36.1	1210	0.7848

Table 1: Best COMET and BLEU scores on EN-CS newstest2020 for all the combinations of models size, training regime and block size. We report the best score and an number of updates after which was this score reached.

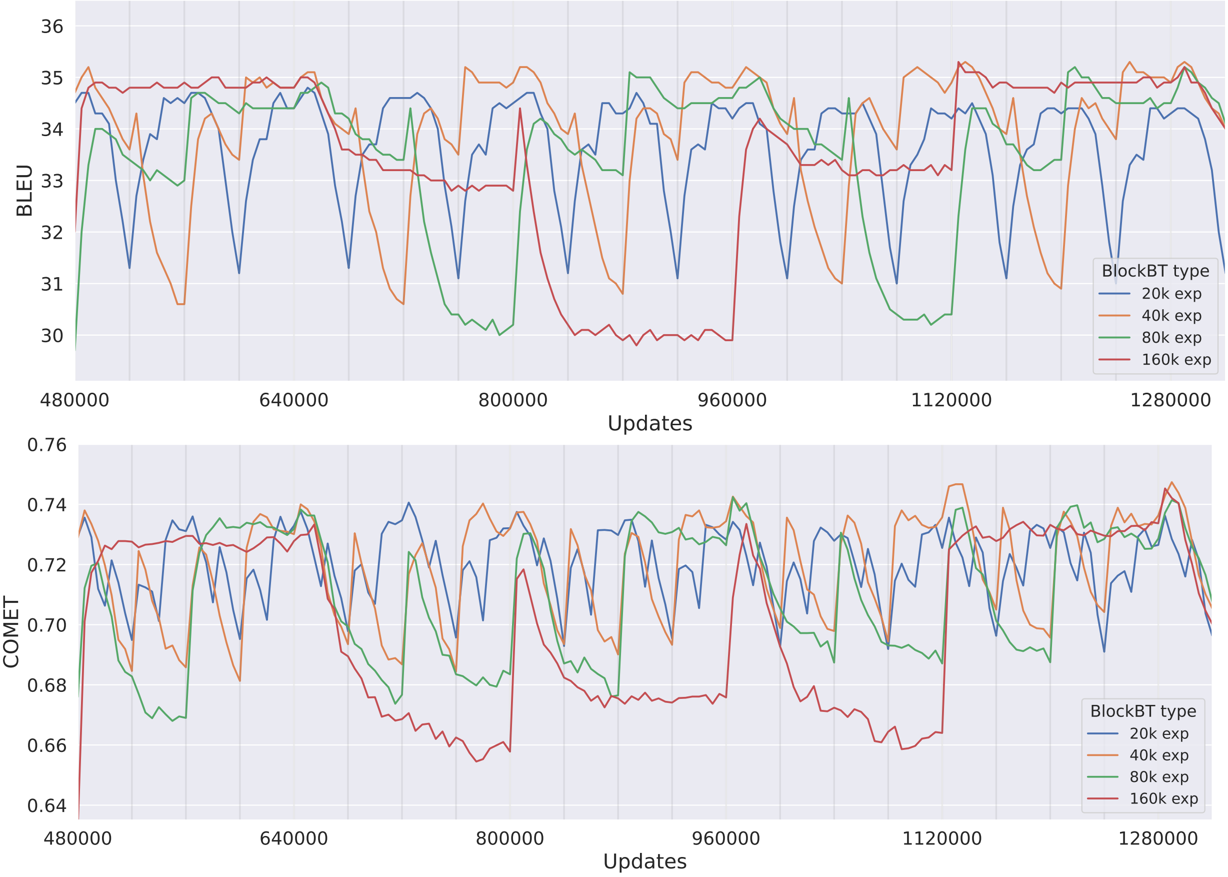


Figure 2: Comparison of how the block size affects behavior of BLEU (top) and COMET (bottom) scores during the training for block-BT with exponential smoothing of the parameters, without checkpoint averaging, on EN-CS `newstest2020`.

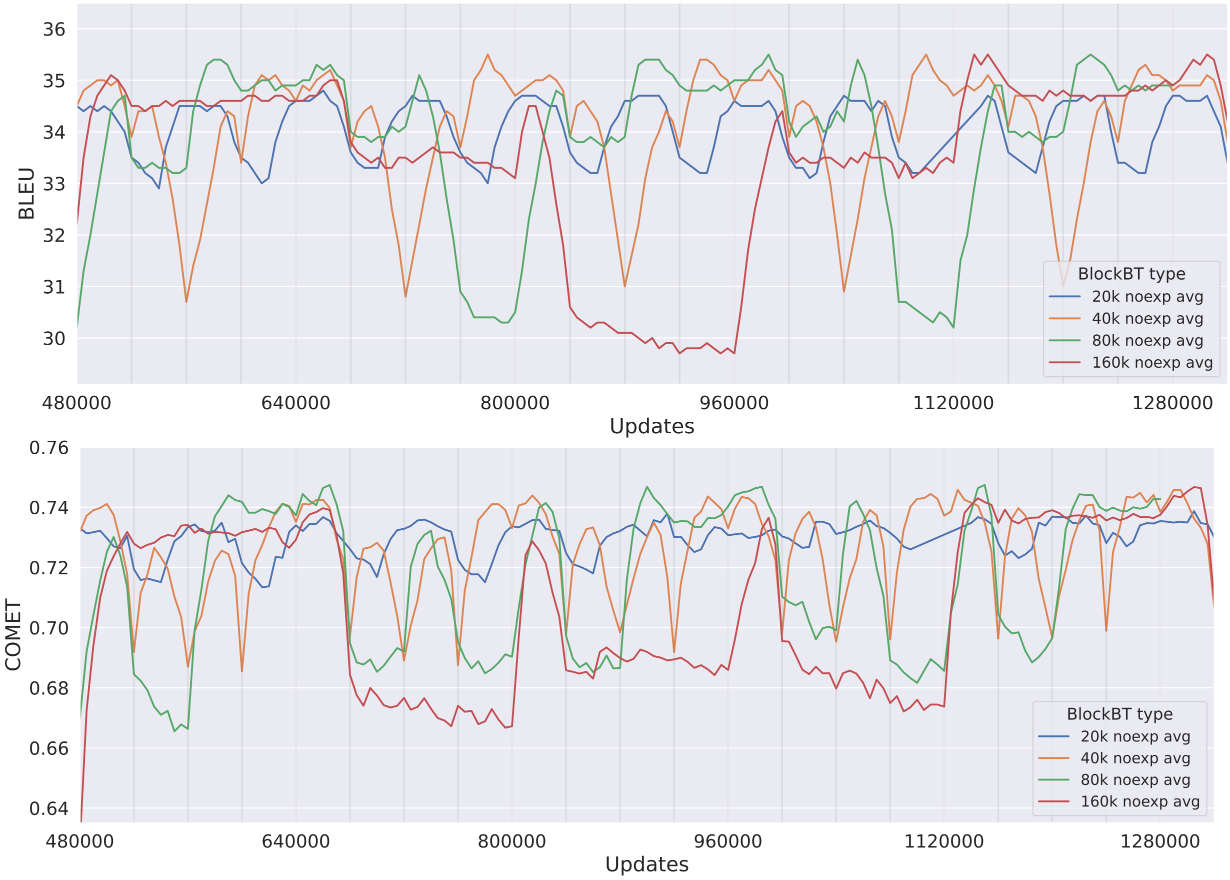


Figure 3: Comparison of how the block size affects behavior of BLEU (top) and COMET (bottom) scores during the training or block-BT with checkpoint averaging and no exponential smoothing of the parameters, on EN-CS newstest2020.

Model	**Block**	**Avg type**	**update (k)**	**best BLEU**	**update (k)**	**best COMET**
base	mixed	exp	1405	25.2	1220	0.4149
		exp+avg8	1430	25.1	1220	0.4114
	40k	-	580	25.3	1040	0.4086
		exp	755	25.3	570	0.4183
		avg8	765	25.4	1060	0.4175
		exp+avg8	1080	25.2	1230	0.4186

Table 2: COMET and BLEU scores for Czech to English directions. The best checkpoints were chosen based on their performance on newstest2020.

dir	regime	datasets	D BLU	T BLU	D CMT	T CMT
encs	mixed	all	34.7	20.9	0.7337	0.6206
		auth+cs	31.5	19.5	0.6904	0.5779
		auth+en	34.8	20.6	0.7258	0.6097
	block	all	35.3	**21.1**	0.7474	**0.6245**
		auth+cs	33.9	19.9	0.7232	0.5908
		auth+en	**35.4**	20.7	**0.7497**	0.6147
csen	mixed	all	25.2	-	0.4149	-
	block	all	25.3	-	0.4183	-
		auth+en	24.3	-	0.3682	-

Table 3: Results on newstest2020 and newstest2021 for various dataset combinations. *D/T* mean dev (*newstest2020*) and test (*newstest2021*) sets respectively, *CMT* stands for wmt20-comet-da scores.

Update (k)	**COMET**	**BLEU**	**Acc**
570	**0.4183**	24.9	0.607
755	0.4038	**25.3**	0.629
590	0.4099	24.9	**0.636**

Table 4: Best checkpoints of Czech to English model trained with 40k blocks and exponential smoothing in terms of COMET (first row), BLEU (second row) on newstest2020 and NE translation accuracy on restaurant test set (third row).

3.6 MBR decoding

We used MBR decoding to rerank concatenation of n-best lists produced by various checkpoints. In total, we used 6-best lists from 12 checkpoints. We divided the checkpoints based on which block of the training data they were saved in and sorted them by COMET score on newstest2020. Using different strategies we selected the best performing checkpoints to provide the n-best lists. We present the results in Table 5. The first row shows results for mixed-BT regime, i.e. we concatenated n-best lists produced by the 12 best performing mixed-BT

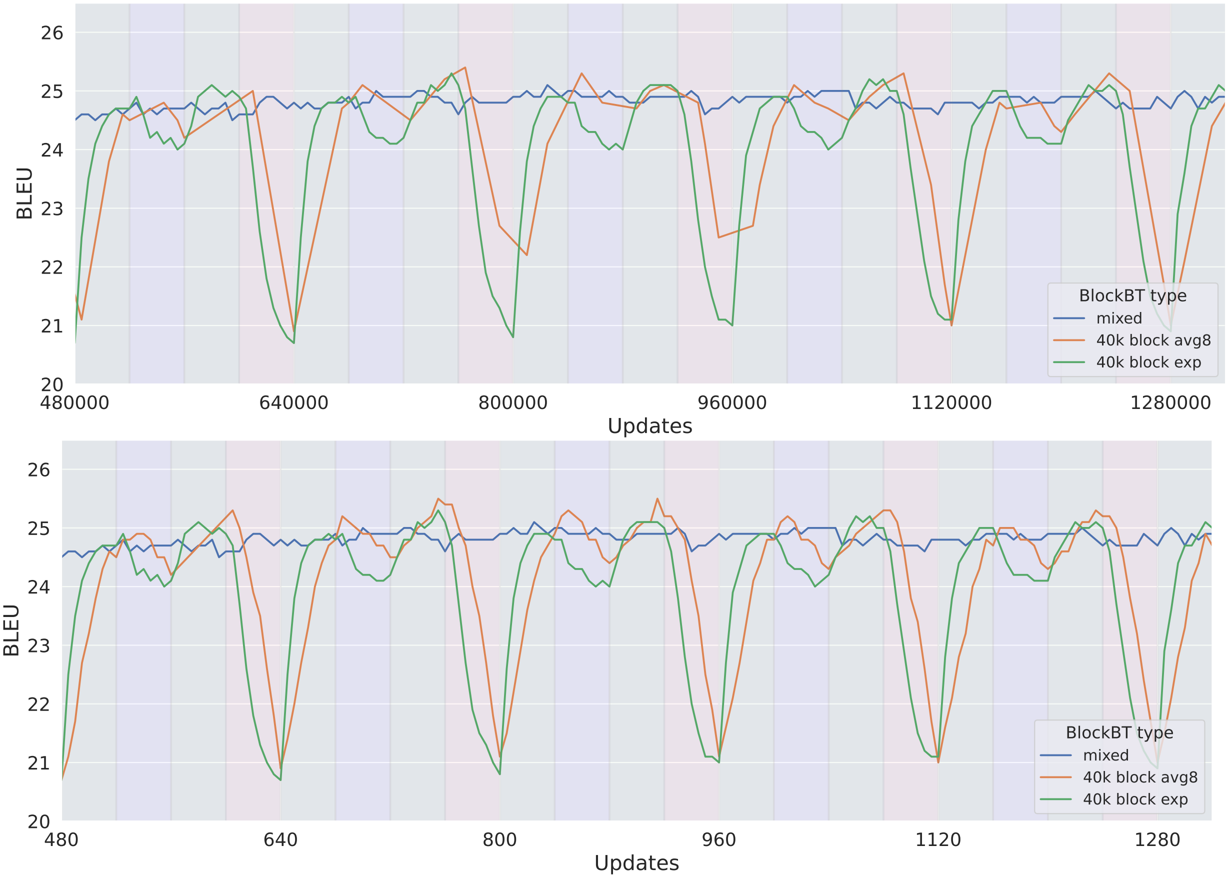


Figure 4: Comparison of different training regimes for CS-EN translation on `newstest2020` in terms of BLEU (top) and COMET (bottom). Background colors for block-BT regime show which part of training data was used for given part of the training. Green means authentic parallel data, blue is CS->EN forward translation and red is EN->CS backtranslation.

i	auth	cs	en	AVG comet20	MBR comet20	comet21
1	-	-	-	0.7322	0.7888	0.0885
2	9	2	1	**0.743**	0.8082	0.0946
3	4	4	4	0.7408	0.8182	0.0972
4	12	0	0	0.7425	0.801	0.0929
5	0	12	0	0.7303	0.8104	0.0949
6	0	0	12	0.7372	0.796	0.0918
7	1	7	4	0.737	**0.8232**	**0.0981**
8	0	7	5	0.7361	**0.8232**	**0.098**
9	2	7	3	0.7377	**0.8231**	**0.0981**

Table 5: Results of MBR decoding on `newstest2020` for different selection of the hypotheses n-best lists produced by checkpoints from different training blocks. In total, 12 n-best lists produced by transformer-base models are concatenated and the first three columns show how many n-best lists are used from each block (the checkpoints for each block are sorted by COMET (wmt20-da model), so these are produced by the best performing checkpoints). The *AVG COMET20* shows the average wmt20-da COMET scores for the first hypotheses of each n-best list that was used, *MBR COMET20* shows wmt20-da score of the final sentences after MBR decoding, COMET21 shows results of the same sentences from wmt21-da model.

checkpoints. In the second row, the block-BT training checkpoints were used to create n-best lists, selected only based on their COMET scores, without any regard on the block type they were saved in. In third row, we combine n-best lists from 4 best performing checkpoints from each type of block. In rows 4-6, we use best performing checkpoints from each type of block separately. In the final row, we show the optimal selection which yielded the highest score. The results suggest that larger diversity in terms of block type of the checkpoints improves MBR results: the combination of n-best lists produced by checkpoints from diverse block types provides a better pool of hypotheses for MBR, even though the average COMET score of these checkpoints is lower than for the less diverse selection. This can be observed in rows 2 and 3.

3.7 Submission

Our primary submission is based on the *big 12-6* model and MBR decoding. We explored all the possible combinations of 18 checkpoints from dif-

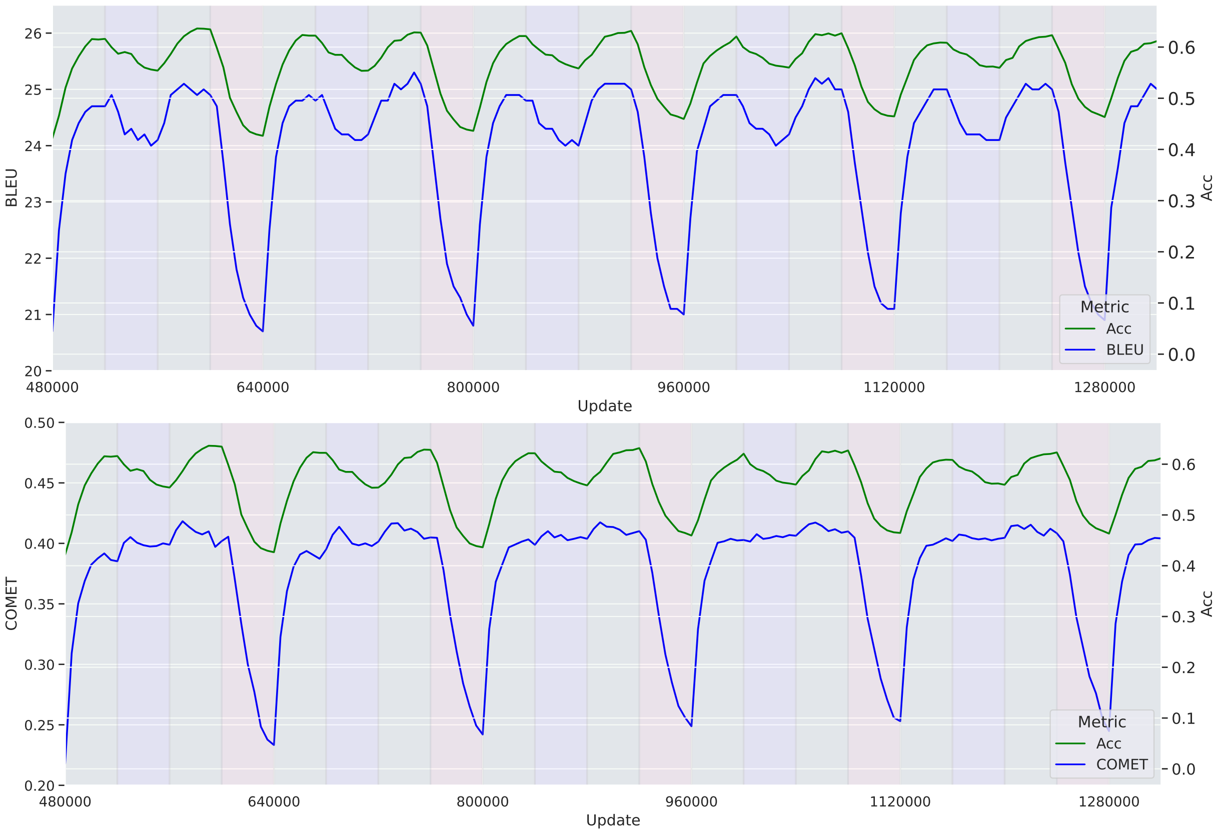


Figure 5: Behaviour of BLEU (top), COMET (bottom) on `newstest2020` and NE translation accuracy on `restaurant` test set for Czech to English translation with block-BT using exponential smoothing.

auth	**cs**	**en**	**AVG comet20**	**MBR comet20**	**comet21**
9	2	8	0.7802	0.8566	0.1114

Table 6: Our final submission for the EN-CS general translation task, based on outputs of the transformer-big 12-6 model. Meaning of the columns is identical to Table 5.

ferent blocks as described in the previous section. The results of the best combination are shown in Table 6. We present the results of the official evaluation in our task in Table 7. In total, there were 5 submitted systems (4 constrained) and 5 online services. Our submission ranked first in COMET score among the constrained systems and third in ChrF score.

System	COMET-B	COMET-C	ChrF-all
Online-W	97.8	79.3	70.4
Online-B	97.5	76.6	71.3
CUNI-Bergamot *	**96.0**	**79.0**	65.1
JDExploreAcademy *	95.3	77.8	**67.2**
Lan-Bridge	94.7	73.8	70.4
Online-A	92.2	71.1	67.5
CUNI-DocTransformer *	91.7	72.2	66.0
CUNI-Transformer *	86.6	68.6	64.2
Online-Y	83.7	62.3	64.5
Online-G	82.3	61.5	64.6

Table 7: Results of automatic metrics on wmt22 general task test set. Constrained submissions are marked by an asterisk, the best scores among constrained submissions are bold. COMET-B and COMET-C are COMET scores for the two different references, ChrF is computed using both references together.

4 Acknowledgements

This work was supported by GAČR EXPRO grant NEUREM3 (19-26934X, RIV: GX19-26934X) and we used services provided by the LINDAT/CLARIAH-CZ Research Infrastructure (`https://lindat.cz`), supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2018101). We would also like to thank Martin Popel for his feedback on the paper and Ondřej Bojar for overall guidance in the field.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Chantal Amrhein and Rico Sennrich. 2022a. Identifying weaknesses in machine translation metrics through minimum bayes risk decoding: A case study for comet.

Chantal Amrhein and Rico Sennrich. 2022b. Identifying weaknesses in machine translation metrics through minimum bayes risk decoding: A case study for comet. *arXiv preprint arXiv:2202.05148*.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Ondřej Bojar and Aleš Tamchyna. 2011. Improving translation model by monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 330–336, Edinburgh, Scotland. Association for Computational Linguistics.

Eikema Bryan and Aziz Wilker. 2021. Sampling-based minimum bayes risk decoding for neural machine translation.

Ondřej Dušek and Filip Jurčíček. 2019. Neural generation for Czech: Data and baselines. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 563–574, Tokyo, Japan. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. 2021. Minimum bayes risk decoding with neural metrics of translation quality.

Petr Gebauer, Ondřej Bojar, Vojtěch Švandelík, and Martin Popel. 2021. CUNI systems in WMT21: Revisiting backtranslation techniques for English-Czech NMT. In *Proceedings of the Sixth Conference on Machine Translation*, pages 123–129, Online. Association for Computational Linguistics.

Vaibhava Goel and William J Byrne. 2000. Minimum bayes-risk automatic speech recognition. *Computer Speech Language*, 14(2):115–135.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *CoRR*, abs/1211.3711.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233, Florence, Italy. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Tom Kocmi, Martin Popel, and Ondrej Bojar. 2020. Announcing czeng 2.0 parallel corpus with over 2 gigawords. *arXiv preprint arXiv:2007.03006*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 169–176, Boston, Massachusetts, USA. Association for Computational Linguistics.

Clara Meister, Ryan Cotterell, and Tim Vieira. 2020. If beam search is the answer, what was the question? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2173–2185, Online. Association for Computational Linguistics.

Mathias Müller and Rico Sennrich. 2021. Understanding the properties of minimum Bayes risk decoding in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint*

Conference on Natural Language Processing (Volume 1: Long Papers), pages 259–272, Online. Association for Computational Linguistics.

Martin Popel. 2018. Machine translation using syntactic analysis.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(4381):1–15.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Felix Stahlberg and Bill Byrne. 2019. On NMT search errors and model errors: Cat got your tongue? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3356–3362, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. Noising and denoising natural language: Diverse backtranslation for grammar correction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 619–628, New Orleans, Louisiana. Association for Computational Linguistics.

Biao Zhang, Ivan Titov, and Rico Sennrich. 2019. Improving deep transformer with depth-scaled initialization and merged attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 898–909, Hong Kong, China. Association for Computational Linguistics.

KYB General Machine Translation Systems for WMT22

Shivam Kalkar, Yoko Matsuzaki, and Ben Li

NRI Digital, Ltd,
{s-kalkar, y-matsuzaki, b4-li}@nri.co.jp

Abstract

We here describe our neural machine translation system for the general machine translation shared task in WMT 2022. Our systems are based on the Transformer (Vaswani et al., 2017) with base settings. We explore the high-efficiency model training strategies, aimed to train a model with high-accuracy by using a small model and a reasonable amount of data. We performed fine-tuning and ensembling with N-best ranking in English to/from Japanese directions. We found that fine-tuning by filtered JParaCrawl data set leads to better translations for both directions in English to/from Japanese models. In the English to Japanese direction model, ensembling and N-best ranking of 10 different checkpoints improved translations. By comparing with another online translation service, we found that our model achieved a great translation quality.

1 Introduction

We participated in the Japanese to/from English translation for the general machine translation shared task of WMT 2022. Japanese ←→English is one of the challenging language pairs for machine translation since their differences are large in both vocabulary and grammatical structure. Recent advances in neural machine translation models have greatly promoted the development of the community. The transformer is the current key model and most recent participants are using a big-setting transformer model to improve the quality of translations. However, developing a more efficient model is also important. We here use a smaller model and limited computation resources to pursue high-quality translation models.

Our systems are based on the Transformer model with base settings, and the models are trained on the parallel corpus of Japanese and English (Morishita et al., 2019). We compared the quality of translations by using fine-tuning with several datasets. Also, we tested several different hyperparameters of the training to find suitable values for the task. After the fine-tuning, we tried to perform ensembling of multiple results from the model to earn a better-quality translation in the English to/from Japanese model. Here we describe the details of our systems.

2 Data selection and preprocessing

We select a suitable parallel corpus for model fine-tuning. We compare WMT provided dataset (which contained 7 different sources including the JParaCrawl dataset), KFTT (Kyoto Free Translation Task data set, Neubig, 2011), the JParaCrawl dataset (ver 2) and so on. We performed fine-tuning for these datasets and found that the model trained on the JParaCrawl dataset achieved better performance. We used a test data set made from WMT provided data and compare model performances by BLEU score. The score of the no fine-tuned model was 37.21, KFTT fine-tuned model was 14.87 and JParaCrawl fine-tuned model was 44.09. Therefore, we decided to use JParaCrawl as our fine-tuning dataset finally. We also consider that JParaCrawl has a reasonable amount of data for our high-efficiency training strategies.

Before we use the dataset, we check the corpus data to clean up. The JParaCrawl dataset contains over 10 million sentence pairs which were constructed by broadly crawling the web and automatically aligning. Therefore, there were noise and low-quality translations. We filtered low quality translation pairs and made a better translation dataset for fine-tuning. We also find that there were some contaminations of non-Japanese languages (e.g., Korean, Chinese) in the Japanese data. We also remove these pairs from the dataset.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 290 - 294
December 7-8, 2022 ©2022 Association for Computational Linguistics

3 Tokenization

We perform the tokenization procedure using the SentencePiece toolkit[1] which provides us with a segmented sentence as tokens. In Japanese and some other languages like Chinese, words were not separated by spaces, therefore, tokenization needs to detect divided positions to separate each token. For Japanese, tokenization can be performed by a lattice-based tokenizer like MeCab[2]. A lattice-based tokenizer performs tokenization based on a dictionary and if the contents of the dictionary cover whole words in data, it provides highly accurate tokenization. However, in the development of machine translation using Neural Network mechanisms, more efficient tokenization methods like Byte-Pair-Encoding (BPE) were proposed (Sennrich et al. 2016c). SentencePiece was developed based on these methods and provides more efficient tokenization for the NMT (Kudo and Richardson, 2018).

SentencePiece is especially effective for languages not using spaces to separate words, has agglutinating morphology, and contains many compound words. Using SentencePiece helps extract subwords within compound words and create a more robust tokenizer. SentencePiece was used again to detokenize by removing the meta symbols from the output translation. For preprocessing the data, we have used the SentencePiece model, in which the vocabulary size is set to 32,000, and sentences whose length exceeded 250 subwords are removed from the training data.

4 Model Training

We train our NMT models with the fairseq[3] toolkit. The models are based on Transformer (Vaswani et al., 2017) with base settings. We use an encoder/decoder with six layers. We set their embedding size to 512, and their feed-forward embedding size to 2048. We use eight attention heads for both the encoder and the decoder. We used dropout with a probability of 0.3. As an optimizer, we used Adam with $\alpha = 0.001$, $\beta1 = 0.9$, and $\beta2 = 0.98$. We used a root-square decay learning rate schedule with a linear warmup of 4000 steps. We clipped gradients to avoid exceeding their norm 1.0 to stabilize the training. For the base settings, each mini-batch contained about 5,000 tokens (subwords), and we accumulated the gradients of 64 mini-batches for updates. We trained the model with 24,000 iterations, saved the model parameters every 200 iterations, and averaged the last eight models. To achieve maximum performance with the latest GPUs, we use mixed-precision training. When decoding, we used a beam search with a size of six as the default condition and length normalization by dividing the scores by their lengths. We test other parameters of a beam search in the model of Japanese → English translations (size = 2, 3, 4, and 10) and found that size = 2 provide the best BLEU score for this task. We also compared models output by scaraBLEU (Post, 2018).

Our models are trained on the Google Cloud Platform's compute engine with 2-T4 GPUs. Model training generally took approximately 3.5 hours. We train our models in mixed precision to save costs without compromising on the accuracy.

Model condition	JParaCrawl data
Pretrained Model	39.4
Finetuned Model	45.1
Finetuned with ensemble	46.9*[1]

Table 1: BLEU Scores of English → Japanese direction, each column uses the same test dataset for three conditions.

*1 This result was not submitted due to our system's trouble.

5 Model Ensembling and N-Best Reranking for English → Japanese direction

After we fine-tuned our base model, we performed model ensembling with N-Best Reranking (Le et. al., 2021). For n-best reranking, we have created a script by referring to a script by Xu Song[4], bert-as-

[1] https://github.com/google/sentencepiece

[2] https://taku910.github.io/mecab/

[3] https://github.com/facebookresearch/fairseq

[4] https://github.com/xu-song/bert-as-language-model

Models	BLEU
Our model	43.9
DeepL	26.6

Table 2 Test result of our model and DeepL

a-language-model. We performed some changes in the scripts for its application to Japanese. For measuring the likelihood of the Japanese sentences produced by the NMT model, we have used the bert-Japanese model released by Yohei Kikuta[5].

For ensembling, the basic idea is to calculate the probability of tokens and perplexity of sentences produced by 10 different checkpoint files of a finetuned model. These 10 checkpoints will create 10 different translations for a given English sentence. Later, we are using bert-as-language-model to calculate the best sentence (the one with the lowest perplexity) score. We have used this sentence output for the submission. This method ensures the selected sentence has maximized fluency compared to other candidates.

6 Results and discussions

6.1 English → Japanese direction

We performed an experiment to compare ensembling effect (Table 1). In the experiment, we prepare training data from the JParaCrawl dataset to fine-tune our model and compare translations with/without ensembling. Based on the same training conditions, the score of the ensembling model is higher than the result of the model without ensembling.

To evaluate our translation quality, we compare the result with the online translation service (DeepL) by using a test dataset which created by the JParaCrawl dataset. The test data contains 1000 sentences that were not contained in the train data. The BLEU score of our model was higher than DeepL this means our fine-tuning procedure leads to better translation for the JParaCrawl dataset (Table 2).

We also check the translation result of the test set released by WMT2022. The dataset consists of 2037 English sentences and there were no reference sentences of Japanese. Therefore, we cannot calculate BLEU score here. Alternatively, we calculate perplexity[6] (PPL), by using bert-japanese model[5], which is explained in the model ensembling section. PPL is a metric of a language model and lower values mean better. We also check the translation quality by the human evaluation of a Japanese native speaker.

The average of the PPL of our model was lower than DeepL (Table 3). The result suggested that our small model established a high-fluently prediction rather than DeepL. In detail, for 941 cases in the test set with 2037 sentences, our PPL was lower than DeepL. We presented several examples of these cases in appendix examples 1 to 4. In these examples, the quality of translations for our model is also better than DeepL based on the confirmation of a native speaker. As a bad case, we list example-5 in the appendix. Although the translation of DeepL has better quality, however, the PPL score was higher than our model's output.

The results above (Table 2 and Table 3) suggested that we can establish a high-quality NMT model by small model and a reasonable amount of data, by using high-efficiency training strategies.

6.2 Japanese → English direction

For the Japanese to English direction, we perform finetuning with the Transformer model base setting on the JParaCrawl dataset. Table 4 shows our training results. For the final submission, we also performed post-processing to delete some extra punctuations that appeared in the translation results. We found that post-processing improved our results by 0.1 BLEU score.

Model condition	Our_PPL	DeepL_PPL	No. of cases
Average	41.59	51.75	2037
Average (our < DeepL)	21.86	90.61	941
Average (our > DeepL)	59.84	15.79	1096

Table 3 Comparison of our model and DeepL outputs by PPL

[5] https://github.com/yoheikikuta/bert-japanese

[6] https://huggingface.co/docs/transformers/perplexity

Model condition	JParaCrawl data
Pretrained Model	37.2
Finetuned Model	44.3

Table 4: BLEU Scores of Japanese to English direction.

7 Conclusions

We explored the high-efficiency model training strategies with a small model and a reasonable amount of data. Our systems are based on the transformer with a base setting. In our experiments, we found that data cleaning, model averaging, model ensembling, beam search, finetuning, parameter-tuning, and post-processing are useful techniques to train a high-quality model. Finally, we compared the translation results between our model and the online translation service, we found that our model achieved better translation quality. Our experiments suggested that exploring more efficient training strategies with a smaller model, a reasonable amount of data, and limited computational resources is promising to achieve a high-quality translation model.

References

Makoto Morishita, Jun Suzuki and Masaaki Nagata. 2019. JParaCrawl: A large scale web-based English-Japanese parallel corpus. *arXiv preprint* arXiv:1911.10668. https://doi.org/10.48550/arXiv.1911.10668

Graham Neubig. 2011. The Kyoto Free Translation Task, http://www.phontron.com/kftt

Rico Sennrich, Barry Haddow and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint* arXiv:1508.07909. https://doi.org/10.48550/arXiv.1508.07909

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics. https://aclanthology.org/D18-2012

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30. https://doi.org/10.48550/arXiv.1706.03762

Giang Le, Shinka Mori, and Lane Schwartz. 2021. Illinois Japanese↔ English News Translation for WMT 2021. In *Proceedings of the Sixth Conference on Machine Translation*, pages 144–153, Online. Association for Computational Linguistics.. https://aclanthology.org/2021.wmt-1.11

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics. https://aclanthology.org/W18-6319

A Appendices

Example1

English: *[Not this time.]*

our_translation（Ja）	our_ppl
″今回はそうではありません。″	3.219
DeepL_translation（Ja）	DeepL_ppl
″今回は違う″	365.825

These two translations are similar, our model translation is a bit better.

Example2

English: *["How are we going to handle this?" he continued.]*

our_translation（Ja）	our_ppl
″どのように私達はこれを処理しようとしているか? 彼は続けた。″	20.209
DeepL_translation（Ja）	DeepL_ppl
″「そして、「この問題にどう対処していくのか？″	84.582

The quality of translations is better for our model based on the confirmation of a native speaker.

Example3

English: *[I have checked and this would be contactless so they would not be able to bring the item to your property I am afraid, I do apologise about this]*

our_translation (Ja)	our_ppl
私はチェックしました、そして、彼らは私が恐れているあなたの財産にアイテムを持って来ることができないので、これは非接触になるでしょう、私はこれについて謝ります。	13.010
DeepL_translation (Ja)	DeepL_ppl
″このような場合、私は、彼らがあなたの財産に項目をもたらすことができないだろう、私はこのことについて謝罪している非接触型であることを確認しました。″	48.198

The quality of translations is better for our model based on the confirmation of a native speaker.

Example4

English: *[If you have any questions, please feel free to contact us through the eBay emailing system.]*

our_translation (Ja)	our_ppl
″ご不明な点がございましたら、Eメールにてお気軽にご連絡ください。″	3.439
DeepL_translation (Ja)	DeepL_ppl
″質問があったら、eBay の emailing システムによって私達に連絡すること自由に感じて下さい。″	15.750

The quality of translations is better for our model based on the confirmation of a native speaker.

Example5

English: *[I've looked into it and I can see that your area is currently having a high volumes of order that is why they were assigning a rider for your order.]*

our_translation (Ja)	our_ppl
″私はそれを調べて、私は、あなたの地域が、現在、それらが、あなたの注文のためにリカーを割り当てていた理由である大量の注文を持っているのを見ることができます。″	18.24
DeepL_translation (Ja)	DeepL_ppl
″調べたところ、あなたの地域では現在注文が集中していて、そのためライダーが割り当てられることになったようです。″	85.75

Although our model PPL is lower, the quality of translations is better for DeepL based on the confirmation of a native speaker.

Analyzing the Use of Influence Functions for Instance-Specific Data Filtering in Neural Machine Translation

Tsz Kin Lam[*]
ICL, Heidelberg University
lam@cl.uni-heidelberg.de

Eva Hasler
Amazon AI Translate
ehasler@amazon.com

Felix Hieber
Amazon AI Translate
fhieber@amazon.com

Abstract

Customer feedback can be an important signal for improving commercial machine translation systems. One solution for fixing specific translation errors is to remove the related erroneous training instances followed by re-training of the machine translation system, which we refer to as instance-specific data filtering. Influence functions (IF) have been shown to be effective in finding such relevant training examples for classification tasks such as image classification, toxic speech detection and entailment task. Given a probing instance, IF find influential training examples by measuring the similarity of the probing instance with a set of training examples in gradient space. In this work, we examine the use of influence functions for Neural Machine Translation (NMT). We propose two effective extensions to a state of the art influence function and demonstrate on the sub-problem of copied training examples that IF can be applied more generally than handcrafted regular expressions.

1 Introduction

Neural Machine Translation (NMT) is the de facto standard for recent high-quality machine translation systems. NMT, however, requires abundant amount of bi-text for supervised training. One common approach to increase the amount of bi-text is via data augmentation (Sennrich et al., 2015; Edunov et al., 2018; He et al., 2019, *inter alia*). Another approach is the use of web-crawled data (Bañón et al., 2020) but since crawled data is known to be notoriously noisy (Khayrallah and Koehn, 2018; Caswell et al., 2020), a plethora of data filtering techniques (Junczys-Dowmunt, 2018; Wang et al., 2018; Ramírez-Sánchez et al., 2020, *inter alia*) have been proposed for retaining a cleaner portion of the bi-text for training.

While standard data filtering techniques aim to improve the quality of the overall training data without targeting the translation quality of specific instances, *instance-specific data filtering* focuses on the improvement of translation quality toward a specific set of input sentences via removal of the related training data. In commercial MT, this selected set of sentences can be the problematic translations reported by customers. One simple approach of instance-specific data filtering in NMT is manual filtering. In manual filtering, human annotators identify translation errors on sentences reported by customer and designs filtering scheme, e.g., regular expressions to search related training examples for removal from the training set.

In this work, we attempt to apply a more automatable technique called influence functions (IF) which is shown to be effective on image classification (Koh and Liang, 2017), and certain NLP tasks such as sentiment analysis, entailment and toxic speech detection (Han et al., 2020; Guo et al., 2020). Given a probing example, influence functions (IF) search for the influential training examples by measuring the similarity of the probing example with a set of training examples in gradient space. Schioppa et al. (2021) use a low-rank approximation of the Hessian to speed up the computation of IF and apply the idea of self-influence to NMT. However, self-influence measures if a training instance is an outlier rather than its similarity with another instance. Akyürek et al. (2022) question the back-tracing ability of IF on the fact-tracing task. They compare IF with heuristics used in Information Retrieval and attribute the worse performance of IF to a problem called *saturation*. Compared to fact-tracing, the target sides of machine translation can be more diverse which complicates the application of IF.

We apply an effective type of IF called *TracIn* (Pruthi et al., 2020) to NMT for instance-specific data filtering and analyze its behaviour by constructing synthetic training examples containing simulated translation errors. In particular, we find

[*] Work done during an internship at Amazon.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 295 - 309
December 7-8, 2022 ©2022 Association for Computational Linguistics

that

- the gradient similarity, also called the influence[1], is highly sensitive to the network component.
- vanilla IF may not be sufficient to achieve good retrieval performance. We proposed two contrastive methods to further improve the performance.
- training examples consisting of copied source sentences have similar gradients even when they are lexically different. This indicates that the use of influence functions can go beyond what can be achieved with regular expressions.
- an effective automation of the instance-specific data filtering remains challenging.

To the best of our knowledge, we are the first to investigate applying IF for instance-specific data filtering to NMT.

2 Method

Influence functions IF is a technique from robust statistics (Hampel, 1974; Cook and Weisberg, 1982, *inter alia*). It aims to trace a model's predictions back to the most responsible training examples without repeated re-training of the model, aka Leave-One-Out. Koh and Liang (2017) extend this idea from robust statistics to deep neural network that requires only the gradient of the loss functions L and Hessian-vector products so that the influence $\mathcal{I}(z, z')$ of two examples z and z' is approximated as

$$\mathcal{I}(z, z') \approx \nabla_\theta L(z')^T H_{\hat{\theta}}^{-1} \nabla_\theta L(z) \quad (1)$$

where $\hat{\theta}$ is the model parameters at optimum and $H_{\hat{\theta}} = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta^2 L(\theta)$ is the Hessian of the model parameters at $\hat{\theta}$. Given n number of training instances and p number of model parameters, the inverse of Hessian has a complexity of $\mathcal{O}(np^2 + p^3)$ which is expensive to compute for deep neural network. There are several proposed methods to speed up the computation of IF, e.g., by computing on a training subset selected by KNN-search (Guo et al., 2020), by approximating the Hessian with LISSA (Agarwal et al., 2017), by computing on a subset of model parameters (Koh and Liang, 2017), or by replacing the Hessian with some other procedures (Pruthi et al., 2020). In this work, we focus on TracIn which is shown to be better than some other variations (Han and Tsvetkov, 2020; Schioppa et al., 2021) in terms of retrieval performance.

TracIn, denoted by $\mathcal{I}_{\text{TracIn}}(z, z')$, replaces the computationally costly Hessian matrix with an identity matrix. The remained gradient dot product, or called the gradient similarity, is instead computed over C number of checkpoints, followed by averaging:

$$\mathcal{I}_{\text{TracIn}}(z, z') = \frac{1}{C}\sum_{i=1}^{C} \nabla_\theta L(z')^T \nabla_\theta L(z) \quad (2)$$

In NMT, given the same source sentence, the magnitude of the gradient in general is positively correlated to the length of the target sentence. In order to reduce the effect of the target length, we normalize equation 2 by the product of $\|\nabla_\theta L(z')\|$ and $\|\nabla_\theta L(z)\|$, or equivalently, we compute the cosine similarity of $\nabla_\theta L(z')$ and $\nabla_\theta L(z)$.

Given a probing instance z' and its probing gradient $\nabla_\theta L(z')$, instances in the training set that yield a positive value of $\mathcal{I}_{\text{TracIn}}(z, z')$ are called the positively influential training instances (+IFTrain) whereas those that yield a negative value of $\mathcal{I}_{\text{TracIn}}(z, z')$ are called the negatively influential training instances (-IFTrain). Taking a gradient step on +IFTrain reduces the loss on the probing example while taking a gradient step on -IFTrain increases it. IF can be used for data filtering by removing the +IFTrain examples of low quality probing samples since their gradients have similar direction. Conversely, if the probing sample is of high quality, removing -IFTrain examples from the training data would be expected to increase translation quality w.r.t. the probing sample.

3 Experimental Setting

Model configuration and training We use Transformer BASE configuration as described in Vaswani et al. (2017) with default setting and implementation in FAIRSEQ. We use a sentence-piece model to create subword units of size 32k. Unless otherwise specified, we pre-trained our NMT on Europarl-v7 data and News Commentary-v12 data in German-English direction from WMT17 for 100 epochs, about 112K updates, using Adam

[1] In this work, we use gradient similarity or influence interchangeably to denote the result of IF. Be aware that TracIn is also one type of IF.

	Samples	Shared parameters		Non-shared parameters			
		∇_{Full}	∇_{Emb}	∇_{srcEmb}	∇_{trgEmb}	∇_{output}	∇_{concat}
Probing	Noch kommt Volkswagen glimpflich durch. Volkswagen gets off lightly.	1	1	1	1	1	1
1	Das £ 1,35 Mrd. teure Projekt soll bis Mai 2017 fertiggestellt werden Volkswagen gets off lightly.	0.153	0.240	0.006	0.287	0.437	0.339
2	Alle in Frage kommenden Produkte wurden aus dem Verkauf gezogen. Volkswagen gets off lightly.	0.238	0.320	0.013	0.230	0.401	0.319
3	Noch kommt Volkswagen glimpflich durch. In 2008, most malware programmes were still focused on sending out adverts.	-0.021	-0.030	-0.149	-0.022	-0.017	-0.040
4	Noch kommt Volkswagen glimpflich durch. We've made a complete turnaround.	-0.007	-0.016	-0.120	-0.003	0.011	-0.013
5	Noch kommt Volkswagen glimpflich durch. Volkswagen gets off lightly!	0.950	0.894	0.973	0.927	0.843	0.873
6	Noch kommt Volkswagen glimpflich durch! Volkswagen gets off lightly.	0.899	0.912	0.873	0.915	0.940	0.927

Table 1: Example showing the changes of influence by network components. Segments that are marked in red are perturbed from the probing example. ∇_X indicates the network components used in computing the influence, ∇_{concat} indicates the concatenation of ∇_{srcEmb}, ∇_{trgEmb} and ∇_{output}.

optimizerion training of 16-bit[2]. The effective mini-batch size is 4096 x 16 tokens and it takes a p3.16xlarge[3] machine on AWS 6 hours for training. We evaluate the MT model on the newstest2017 test set with a checkpoint averaged over the 10-best checkpoints, measured by the validation loss on the newstest2014-2016 dev set. On the test set, our NMT model with non-shared parameters with the two word embeddings and the output layer scores 29.99 BLEU whereas the one with shared parameters scores 29.78 BLEU. We use beam search with beam size of 5 in decoding.

TracIn We select 5 checkpoints, i.e., at epoch 5, 8, 15, 30 and 100 for computing TracIn[4]. We select checkpoints which have relatively large changes in the validation loss, i.e., usually in the earlier phrase of training, and include the last one to cover information at the end of the training. We compute the per-sample gradient with a batch size of 1 parallelized over multiple processes with several g4dn.2x[3] machines on AWS.

4 Experimental results

This section describes our findings on the properties of applying IF on NMT for instance-specific data filtering.

4.1 Sensitivity of gradient similarity to the network components

In previous works, the influence, or called the gradient similarity, is usually computed with respect to a small part of the network parameters, especially the last or the last few layers (Han et al. (2020);Barshan et al. (2020); *inter alia*). In NMT, we found that the resulting influence is highly sensitive to the network components used in computing the gradients (or gradient component). For illustration, we construct a set of perturbed instances, compute its influence by different gradient components and observe their changes. The perturbed instances are not included during the NMT training. This independence between the NMT and the perturbed instances provides a simpler setting for checking how gradient components and the perturbed examples affect the influence.

[2] We use 32-bit precision to compute the gradient similarity once the training is done.

[3] See https://aws.amazon.com/ec2/instance-types/ for details.

[4] It is tempting to just use the deployed checkpoint to compute the influence. As shown by Liang et al. 2017, however, the Hessian term in equation 1 captures more accurately the effect of model training than the dot product of the optimal checkpoint. In TracIn, the Hessian is approximated by the average over a set of checkpoints, and we follow their guidelines for checkpoints selection.

Table 1 shows the gradient similarities of a probing example from newstest2017 with six artificially created instances. We use two NMT models, 1) trained with shared parameters between the two word embeddings and the output layer and 2) trained without parameter sharing, to compute the similarities.

We notice that gradient similarity for the model with shared parameters is more strongly influenced by lexical matches on the target side, as shown by the larger magnitude of influence values for probing examples 1 and 2 with random source sides compared to probing examples 3 and 4 with random target sides. For non-shared parameters, we observe that the gradient w.r.t. the output layer (∇_{output}) has stronger response (0.437 and 0.401) to the probing instances with random source side whereas the gradient w.r.t. source embedding (∇_{srcEmb}) has stronger response (-0.149 and -0.120) to the instances with random target sides. On the same probing example, we repeat this random sampling of source and target sentences by using the other 3003 instances in the newstest2017 set. We find that the mean magnitude of ∇_{srcEmb} is 0.04 for random target whereas it is 0.004 for random source. In the case of ∇_{output}, the mean magnitude for random target is 0.021 whereas it is 0.428 for random source. This indicates that ∇_{output} has a tendency of scoring sentence pairs higher when their target side overlaps with the target side of the probing instance and is less influenced by source-side overlap. This may be suboptimal for retrieving problematic training examples that are relevant to a given probing instance.

When using a gradient vector ∇_{concat} which is the concatenation of ∇_{srcEmb}, ∇_{trgEmb} and ∇_{output}, its similarity is dominated by ∇_{output} rather than equally shared between the three given that they have the same number of parameters. This may explain why, in the case of shared parameters, instances with random source side have higher similarities than those with random target side.

Instance 5 and 6 are minor edits of the probing instance with changes to punctuation. For instance 5, it is not easy to interpret the results for the model with shared parameters. However, in the non-shared parameter setting, we observe a higher similarity for ∇_{srcEmb} than for ∇_{trgEmb} and ∇_{output}. This is more interpretable because the punctuation change is on the target side. For instance 6, the punctuation change is on the source side and we see a higher TracIn value for ∇_{output} than for ∇_{srcEmb} and ∇_{trgEmb}. As before, the value of ∇_{concat} is more similar to the value of ∇_{output}. Further examples can be found in Table A1 in the Appendix.

These qualitative results show that the choice of network component is crucial in computing the gradient similarity. As shown in the next experiment, this affects the retrieval of training examples.

4.2 Contrastive signal is crucial for better retrieval performance

In this section, we try to illustrate how different gradient components affect the retrieval of the noisy instances with TracIn. We add control to the retrieval outcome by adding synthetic noisy training instances to the training data. In addition, we show that vanilla IF may not be sufficient to achieve good performance because the gradients are aggregated over all tokens in the target sentence. We thus propose two contrastive methods to sharpen the gradient signal.

Synthetic noisy examples We use the error template *X → Y* which stands for *X is translated to Y* to construct synthetic noise examples for the training set . We created four simple error patterns: 1) *August → January*, 2) *Deutschland → Italy*, 3) *Oktober → December* and 4) *Türkei → New Zealand.*

Error pattern	Number of instances		
	train	synthetic noisy	probing
August → January	8,017	925	9
Deutschland → Italy	15,360	4,891	30
Oktober → December	11,927	2,422	8
Türkei → New Zealand	14,963	7,417	22

Table 2: Number of instances per error pattern

In the training set, we replace the translation of the sentences containing the source pattern by the erroneous translation with a probability of 60% so that the total number of training data is unchanged. We select these error patterns because translation errors of months and country names can easily result from noisy training examples and are therefore suitable to simulate real customer issues. In addition, there are related source sentences in the test set, i.e., newstest2017, which can be used as probing examples. In order to speed up the computation of IF, we extract a subset of training data containing the original pattern, the perturbed pattern and some randomly sampled training sentences. For example, in the error pattern *Oktober* → *December*, the training subset contains sentences with *Oktober*, *Dezember*, *October* and *December* on either the source or target side together with some randomly sampled sentences. Table 2 gives the exact number of instances for each case. We follow the same training procedure as section 3 to pre-train a NMT model on the training corpus perturbed by the synthetic noises.

Contrastive-IF The gradient of a source-target pair in NMT involves complicated mapping between the source tokens and the target tokens. That is, the gradient vector does not just contain the information of the error pattern but also other context. In order to isolate the gradient of the error pattern from the aggregated signal, we propose two methods: 1) gradient masking and 2) gradient difference. Both methods leverage a cleaner translation either in the form of a gold-reference translation or a corrected hypothesis, i.e. the hypothesis with the error pattern corrected. We refer to them as *Contrastive Influence Functions* (Contrastive-IF).

The idea of gradient masking (*Mask*) is to apply a 0/1 token-level mask to the loss function so as to remove the contribution of irrelevant tokens from the gradient computation. We assign the mask based on which tokens differ between hypothesis and reference. If the 0-mask is applied everywhere except for the location of the error according to a corrected translation, we refer to it as *MaskExact*.

We can use the difference between two hypotheses in a continuous fashion by simply subtracting their gradients. Specifically, we compute the difference of the gradient of a sentence A and the gradient of a sentence B as the probing gradient: $GD(A, B) = \nabla(A) - \nabla(B)$. In this work, we use the hypothesis as A and a cleaner translation as B (either the reference or the corrected hypothesis) so that positively influential training instances w.r.t. to $GD(A, B)$ are the synthetic noisy training instances.

Results Table 3 shows the retrieval performance of vanilla IF, gradient masking and gradient difference where the gradient is computed w.r.t. to either the source embedding, output layer or the full model. We evaluate the performance with precision over the top-X% influential training instances, i.e. the number of synthetic training instances successfully retrieved given top-X% of the influential training samples. We combine results of the four error patterns by (macro) averaging their precision.

The first three rows show results for vanilla IF (TracIn) when either the hypothesis, the reference or a corrected hypothesis is used for probing the training data. Using ∇_{srcEmb} or ∇_{output} obtain substantially higher precision for each variant than using ∇_{Full}, i.e., the gradient w.r.t. the entire model, which demonstrates the importance of the choice of gradient component(s) in vanilla-IF for retrieval performance. Using the corrected hypotheses to retrieve negatively-influential examples yields the best precision for both top-1% and top-10% of retrieved training examples.

We qualitatively examine the influential instances retrieved. By using the source-hypothesis pair as the probing instance, we find that instances retrieved via ∇_{output} have less similarity on the source side. In the first probing example, *Januar* → *January* occurs more frequently in the ranking than *August* →*January*. In the second example, *Italien* → *Italy* appears as the third influential training instance when using ∇_{output} whereas all top-3 influential instances obtained by ∇_{srcEmb} contain the desired error pattern of *Deutschland* → *Italy*, see Table A2 in the Appendix.

We find that both gradient masking, ∇(HYP$_{\text{Mask}}$), and gradient difference, ∇(HYP) − ∇(REF), perform better than the vanilla IF given the same gradient component. ∇(HYP$_{\text{Mask}}$) always outperforms the comparable vanilla IF variants ∇(HYP) and ∇(REF). If we can identify the exact location of the error pattern, with the probing gradient ∇(HYP$_{\text{MaskExact}}$) or ∇(CorrHYP$_{\text{MaskExact}}$), the precision can be further boosted and this is consistent for gradients ∇_{srcEmb}, ∇_{output} and ∇_{Full}. While the gradient difference variants do not always outperform the comparable masking variants for all ∇_X, ∇(HYP) − ∇(CorrHYP) yields the overall best result using ∇_{srcEmb}.

An interesting finding is the improvement brought by the corrected hypothesis (CorrHYP). Applying vanilla-IF on it already achieves a precision of 0.930 under ∇_{srcEmb} considering the top-1% influential instances. By applying *MaskExact* or gradient difference on it, we achieve very high precisions of 0.989 and 1.0 under ∇_{srcEmb} considering the top-1% influential training instances. One notable gain brought by the proposed approaches is that for ∇_{Full}, the precision increases from 0.531 to around 0.987 for the ∇(HYP) − ∇(CorrHYP) variant, bringing it on-par to the performance of

∇(Probing)	+/-	Precision		
		∇_{srcEmb}	∇_{output}	∇_{Full}
∇(HYP)	+	0.846	0.720	0.503
∇(REF)	-	0.876	0.794	0.481
∇(CorrHYP)	-	0.930	0.905	0.531
∇(HYP$_{\text{Mask}}$)	+	0.893	0.840	0.654
∇(HYP$_{\text{MaskExact}}$)	+	0.957	0.910	0.862
∇(CorrHYP$_{\text{MaskExact}}$)	-	0.989	**0.992**	0.924
∇(HYP) - ∇(REF)	+	0.930	0.856	0.584
∇(HYP) - ∇(CorrHYP)	+	**1.000**	0.971	**0.987**

(a) Retrieval performance for top-1% influential training examples

∇(Probing)	+/-	Precision		
		∇_{srcEmb}	∇_{output}	∇_{Full}
∇(HYP)	+	0.765	0.644	0.442
∇(REF)	-	0.799	0.693	0.437
∇(CorrHYP)	-	0.844	0.781	0.455
∇(HYP$_{\text{Mask}}$)	+	0.848	0.829	0.567
∇(HYP$_{\text{MaskExact}}$)	+	0.936	0.904	0.825
∇(CorrHYP$_{\text{MaskExact}}$)	-	0.962	**0.958**	0.875
∇(HYP) - ∇(REF)	+	0.855	0.764	0.515
∇(HYP) - ∇(CorrHYP)	+	**0.986**	0.935	**0.931**

(b) Retrieval performance for top-10% influential training examples

Table 3: Retrieval performance measured in (macro) averaged precision over all error patterns. $\nabla(Probing)$ refers to the gradient with input 'source-Probing'. HYP, REF and CorrHYP stands for hypothesis, reference and corrected hypothesis respectively. "+" ("-") indicates that positively (negatively) influential training instances were retrieved. ∇_X indicates network components used in computing the gradient. We mark the best result per column in bold.

∇_{output}. We include results for additional gradient components in Table A3 in the Appendix.

∇(Probing)	top-X% influential training samples	+/-	Precision	
			∇_{Emb}	∇_{Full}
∇(HYP)	1%	+	0.660	0.502
	10%		0.596	0.444
∇(CorrHYP)	1%	-	0.877	0.541
	10%		0.746	0.463
∇(HYP) - ∇(CorrHYP)	1%	+	0.891	0.691
	10%		0.808	0.607

Table 4: Retrieval performance measured in average precision across all error patterns for an NMT model with shared parameters between the word embeddings and the output layer.

We also conducted a side experiment with a NMT model with shared parameters between the embeddings and the output layer. Similar to the case of a NMT model with non-shared parameters, gradient difference improves over the vanilla-IF when averaging precisions over all error patterns as shown in Table 4.

To summarize, both our contrastive-IF variants improve retrieval performance regardless of the network component used in computing gradients and whether the NMT model has shared parameters.

4.3 Copied source sentences have similar gradient signature

Our initial motivation for applying influence functions to NMT was to arrive at a more automatable way of retrieving relevant training examples for reported translation problems. We were also hoping to generalize over what can be achieved by applying manually composed regular expressions which are limited to detecting lexical overlap. In this section, we focus on the latter and investigate whether Influence Functions can retrieve training examples that cause an undesired copy behaviour in the decoder.

Experimental settings On top-of the Europarl-v7 and News Commentary-v12 data, we append a set of 176,004 copied source sentences provided by Khayrallah and Koehn (2018) to the training set. Following the training recipe in section 3, our NMT with non-shared parameters has a degradation of translation quality from 29.99 BLEU to

∇(Probing)	+/-	Precision		
		∇_{srcEmb}	$\nabla_{encoder}$	∇_{Full}
∇(HYP)	+	0.930	0.972	0.994
∇(REF)	-	0.525	0.452	0.548
∇(HYP) - ∇(REF)	+	0.708	0.712	0.949

(a) Retrieval performance for top-10% influential training examples

∇(Probing)	+/-	Precision		
		∇_{srcEmb}	$\nabla_{encoder}$	∇_{Full}
∇(HYP)	+	0.888	0.932	0.986
∇(REF)	-	0.508	0.449	0.504
∇(HYP) - ∇(REF)	+	0.670	0.647	0.895

(b) Retrieval performance for top-20% influential training examples

Table 5: Retrieval performance measured in averaged precision over the probing instances, on copied training instances. $\nabla(Probing)$ refers to the gradient with input 'source-Probing'. HYP, REF stands for hypothesis, reference. "+" ("-") indicates that positively (negatively) influential training instances were retrieved. ∇_X indicates the network components used in computing the gradient.

17.64 BLEU on the newstest2017 data, showing the detrimental effect of the untranslated target sides.

We select 40 probing instances from the newstest2017 data where their translation by the above NMT model is a copy of the source sentence. We again reduce the computation time by running TracIn over a training subset which contains the newly added noisy data, i.e., 176,004 instances and a set of randomly sampled training instances. This creates a training subset of 476,004 instances.

Results Table 5 shows the retrieval performance on copied source sentences in the training subset with probing gradients of $\nabla(HYP)$, $\nabla(REF)$ and $\nabla(HYP)$ - $\nabla(REF)$ computed over source embedding (∇_{srcEmb}), the encoder ($\nabla_{encoder}$), or the entire model (∇_{Full}). We skip the masking strategy in this case since it would mask all target tokens, resulting in a loss of 0. Different from our results so far, the vanilla IF using only the hypothesis preforms better than using the reference for retrieval and better than the gradient difference variant for all network components. For example, when considering only the top-10% influential training instances, the precision is 0.930 for $\nabla(HYP)$ with ∇_{srcEmb} and only 0.525 for $\nabla(REF)$. This may indicate that instances of copied source sentence have similar gradient signature despite their lexical difference (see Table A4 for some examples) and that the reference translation is less useful in this setting because it cannot provide a specific contrastive signal.

A surprising finding in this setting is that using gradients computed over the entire network is better than the source embedding or the entire encoder. This is in contrast to the previous findings in the synthetic training instances. This possibly indicates that the copy mechanism is spread over the entire model or parts beyond the source embedding or the encoder.

4.4 An effective IF-based instance-specific data filtering is hard to automate

Many data filtering algorithms require a threshold to decide which instances are to be filtered. This threshold can be a model score in an offline filtering algorithm (Junczys-Dowmunt, 2018) or a dynamic formula that is changed according to the learning state of the model (Wang et al., 2018). In both cases, a desirable threshold should be effective as measured in the downstream model performance and be easily computed and generalized to other situations. In the case of IF-based instance-specific data filtering, we observe two properties in the ranking of the influence which makes the automation of the data filtering algorithm challenging.

1: The range of influence varies across probing examples Although the influence is bounded between $[-1, 1]$ because of the cosine similarity, the maximum magnitude of the influence for each probing example can still be very different. Table 6 shows the mean and standard deviation of the maximum influence value of positively influential training instances computed over probing examples of the same configuration. Firstly, the mean value is quite diverse across different gradient components, and across different probing gradients of the same error pattern. For example, the mean value of the error pattern *August* → *January* computed with ∇_{srcEmb} is 0.399 or 0.059 depending on which probing gradient is used. Secondly, the standard deviation within each configuration is relatively large when compared to the corresponding mean value. For example, it is about 26%, 36%, 22% and 19% in the case of ∇_{srcEmb} using gradient difference as the probing gradient. This large standard deviation indicates the difficulty of setting an effective threshold for filtering even for probing examples with the same type of error pattern.

Error pattern	∇(HYP) - ∇(CorrHYP)		∇(HYP)	
	∇_{srcEmb}	∇_{Full}	∇_{srcEmb}	∇_{Full}
August → January	0.399 ± 0.104	0.199 ± 0.041	0.059 ± 0.023	0.119 ± 0.042
Oktober → December	0.524 ± 0.192	0.397 ± 0.123	0.056 ± 0.028	0.143 ± 0.043
Deutschland → Italy	0.576 ± 0.126	0.428 ± 0.047	0.097 ± 0.061	0.135 ± 0.046
Türkei → New Zealand	0.527 ± 0.100	0.540 ± 0.118	0.080 ± 0.044	0.165 ± 0.051

Table 6: Statistics showing the mean and standard deviation of the largest influence per configuration. The large standard deviation of the maximum influence value for probing examples of the same error pattern shows the difficulty of defining a comparable filtering threshold across probing instances.

Error pattern	∇(HYP) - ∇(CorrHYP)		∇(HYP)	
	∇_{srcEmb}	∇_{Full}	∇_{srcEmb}	∇_{Full}
August → January	1.44 ± 0.50	3.33 ± 1.76	1.78 ± 1.55	1.44 ± 0.69
Oktober → December	2.25 ± 0.43	2.00 ± 0.00	2.88 ± 1.76	2.00 ± 1.58
Deutschland → Italy	1.00 ± 0.00	1.77 ± 0.62	1.67 ± 1.22	2.70 ± 2.62
Turkei → New Zealand	3.05 ± 1.46	1.32 ± 1.26	2.27 ± 2.09	2.32 ± 1.66

Table 7: Mean and standard deviation of the number of influential training instances to be removed per configuration, using the largest consecutive difference found in the ranking as clustering criterion.

2: The influence value drops abruptly at the top-of the ranking Apart from a fixed threshold across different probing example, we also examine the possibility of automatically setting a threshold for each probing example.

We first examine a simple clustering strategy by searching for the position where the consecutive difference is the largest in the ranking of influence. Table 7 shows the result of the mean and standard deviation of the number of most influential training instances to be removed per configuration. By considering only the largest consecutive difference, less than 5 training instances would be removed which is far less than the number of synthetic training instances.

We examine further by investigating the shape of the influence of the positively influential training instances in the ranking. Figure 1 shows the influences, computed via TracIn, of the top-500 positively influential training instances per error pattern. For each error pattern, we randomly select a probing example to examine its influence under different gradient conditions. In all these cases, the influence drops sharply in the first few instances, especially in the case of vanilla IF, denoted by “GradHYP” in the figures. After the sharp drop, the influence becomes quite steady for the remaining instances. This steady behaviour holds even for instances of much lower rank, see Figure A1 in the Appendix. The “elbow” occurs before the first 50 influential training instances, which includes only a tiny portion of the synthetic noisy training instances.

How about Top-K filtering? In previous work, the authors use either Top-K or Top-X% as the filtering threshold which is not realistic in the case of NMT where 1) there can be billions of training instances, and 2) the error types are more diverse than the prediction of wrong classes. In spite of the good retrieval performance demonstrated in the previous section, our results here show that an effective automation of the IF-based instance-specific data filtering for NMT remains a challenge.

5 Conclusion

We have analyzed the use of Influence Functions for NMT as instance-specific data filtering. By constructing synthetic instances, we found that 1) the gradient similarity is very sensitive to the selected network components, 2) vanilla Influence Functions are not sufficient for good retrieval performance, 3) our proposed contrastive-IF can boost the retrieval performance regardless of the gradient component or parameter sharing, 4) finding an effective automation of IF for instance-specific data filtering is difficult. This is because the proper choice of gradient component with respect to the type of error in the probing example is crucial for the effectiveness of Influence Functions. Despite the reported effectiveness for certain classification tasks in previous literature, our results show that applying IF to NMT poses some practical difficulties that we have not yet been able to solve.

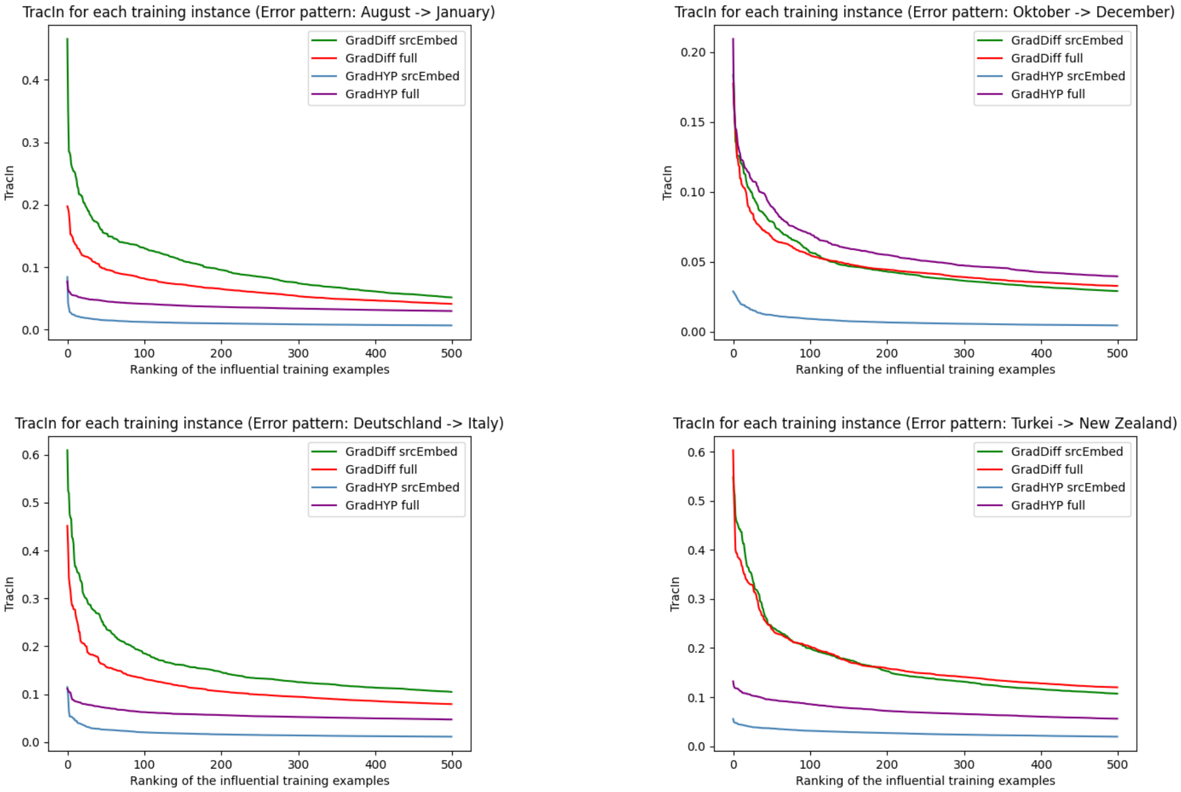


Figure 1: TracIn of the top-500 positively influential training examples. In each subfigure, we randomly select a probing example from each error pattern to compute its influence using gradient difference w.r.t. 1) source embedding (GradDiff srcEmbed), 2) entire model (GradDiff full) and using vanilla-IF with source-hypothesis as input w.r.t. 1) source embedding (GradHYP srcEmbed), 2) entire model (GradHYP full).

6 Limitations

In this work, we provided an analysis of using Influence Functions for Neural Machine Translation as instance-specific data filtering for the purpose of cost saving and finding a more generally applicable solution. Despite the reported success of some previous works in NLP/Vision-related classification tasks, we faced several challenges in applying Influence Functions to NMT. We are aware of the following limitations to our analysis:

- Our analysis focuses on TracIn rather than other influence functions because TracIn is reported to be very effective.
- Our analysis is based on a fixed set of checkpoints, following the practice of previous works. The selection and the number of checkpoints used in TracIn are computationally costly hyper-parameters.
- Our analysis focuses on major network components such as embeddings, encoder and the output layer, excluding other possible combinations.
- The scale of our experiments is limited, e.g., only the De-En language direction with 3M training instances and the synthetic examples are relatively simple. However, given such simple setting, we can already see the challenges of applying IF on NMT as instance-specific data filtering or as an attribution/interpretable method.
- The proposed contrastive IF requires a corrected translation, e.g., reference translation.

We hope that our analysis can inspire further evaluation and modification of the technique.

Acknowledgements

We sincerely thank Ke Tran, Tobias Domhan, Bill Byrne and Sony Trenous for fruitful discussions and support.

References

Naman Agarwal, Brian Bullins, and Elad Hazan. 2017. Second-order stochastic optimization for machine learning in linear time. *The Journal of Machine Learning Research*, 18(1):4148–4187.

Ekin Akyürek, Tolga Bolukbasi, Frederick Liu, Binbin Xiong, Ian Tenney, Jacob Andreas, and Kelvin Guu. 2022. Tracing knowledge in language models back to the training data. In *arXiv preprint arXiv: 2205.11482*.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, et al. 2020. Paracrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567.

Elnaz Barshan, Marc-Etienne Brunet, and Gintare Karolina Dziugaite. 2020. Relatif: Identifying explanatory training samples via relative influence. In *International Conference on Artificial Intelligence and Statistics*, pages 1899–1909. PMLR.

Isaac Caswell, Theresa Breiner, Daan van Esch, and Ankur Bapna. 2020. Language id in the wild: Unexpected challenges on the path to a thousand-language web text corpus. *arXiv preprint arXiv:2010.14571*.

R Dennis Cook and Sanford Weisberg. 1982. *Residuals and influence in regression*. New York: Chapman and Hall.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Han Guo, Nazneen Fatema Rajani, Peter Hase, Mohit Bansal, and Caiming Xiong. 2020. Fastif: Scalable influence functions for efficient model interpretation and debugging. *arXiv preprint arXiv:2012.15781*.

Frank R Hampel. 1974. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393.

Xiaochuang Han and Yulia Tsvetkov. 2020. Fortifying toxic speech detectors against veiled toxicity. *arXiv preprint arXiv:2010.03154*.

Xiaochuang Han, Byron C Wallace, and Yulia Tsvetkov. 2020. Explaining black box predictions and unveiling data artifacts through influence functions. *arXiv preprint arXiv:2005.06676*.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. *arXiv preprint arXiv:1909.13788*.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. *arXiv preprint arXiv:1809.00197*.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. *arXiv preprint arXiv:1805.12282*.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33:19920–19930.

Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón, and Sergio Ortiz-Rojas. 2020. Bifixer and bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298.

Andrea Schioppa, Polina Zablotskaia, David Vilar, and Artem Sokolov. 2021. Scaling up influence functions. *arXiv preprint arXiv:2112.03052*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018. Denoising neural machine translation training with trusted data and online data selection. *arXiv preprint arXiv:1809.00068*.

A Appendix

	Samples	∇_{Full}	∇_{Emb}	∇_{srcEmb}	∇_{trgEmb}	∇_{output}	∇_{concat}
Probing	Selbst die britische Queen hat ihn schon geadelt. Even the British Queen has bestowed an honour upon him.	1	1	1	1	1	1
1	Nur fehlten die Beweise. Even the British Queen has bestowed an honour upon him.	0.358	0.284	0.024	0.225	0.401	0.319
2	Biologen haben in Hannover untersucht, welchen Effekt das Rufen von Katzenbabys auf erwachsene Tiere hat. Even the British Queen has bestowed an honour upon him.	0.275	0.168	0.004	0.219	0.280	0.200
3	Selbst die britische Queen hat ihn schon geadelt. The German branch of the Gülen movement also fears that many Turks will flee abroad.	-0.035	-0.038	-0.125	0.025	-0.043	-0.036
4	Selbst die britische Queen hat ihn schon geadelt. Demonstrators demanding political change in Ethiopia have been met with violent resistance by the government.	-0.039	-0.013	-0.141	0.039	0.001	-0.003
5	Selbst die britische Queen hat ihn schon geadelt. Even the British Queen has bestowed an honour upon him!	0.962	0.924	0.992	0.981	0.905	0.924
6	Selbst die britische Queen hat ihn schon geadelt! Even the British Queen has bestowed an honour upon him.	0.908	0.899	0.912	0.949	0.935	0.935

Table A1: Another example showing the changes of gradient similarity by selected network components. Segments that are marked in red are perturbed from the probing example. The notation ∇_X indicates the network components used in computing the gradient similarity. ∇_{srcEmb} has a mean magnitude of 0.051 and 0.007 on random target and random source respectively whereas ∇_{output} has respectively a mean magnitude of 0.0145 and 0.350. This shows that ∇_{output} has a tendency of scoring sentence-pairs containing random source higher.

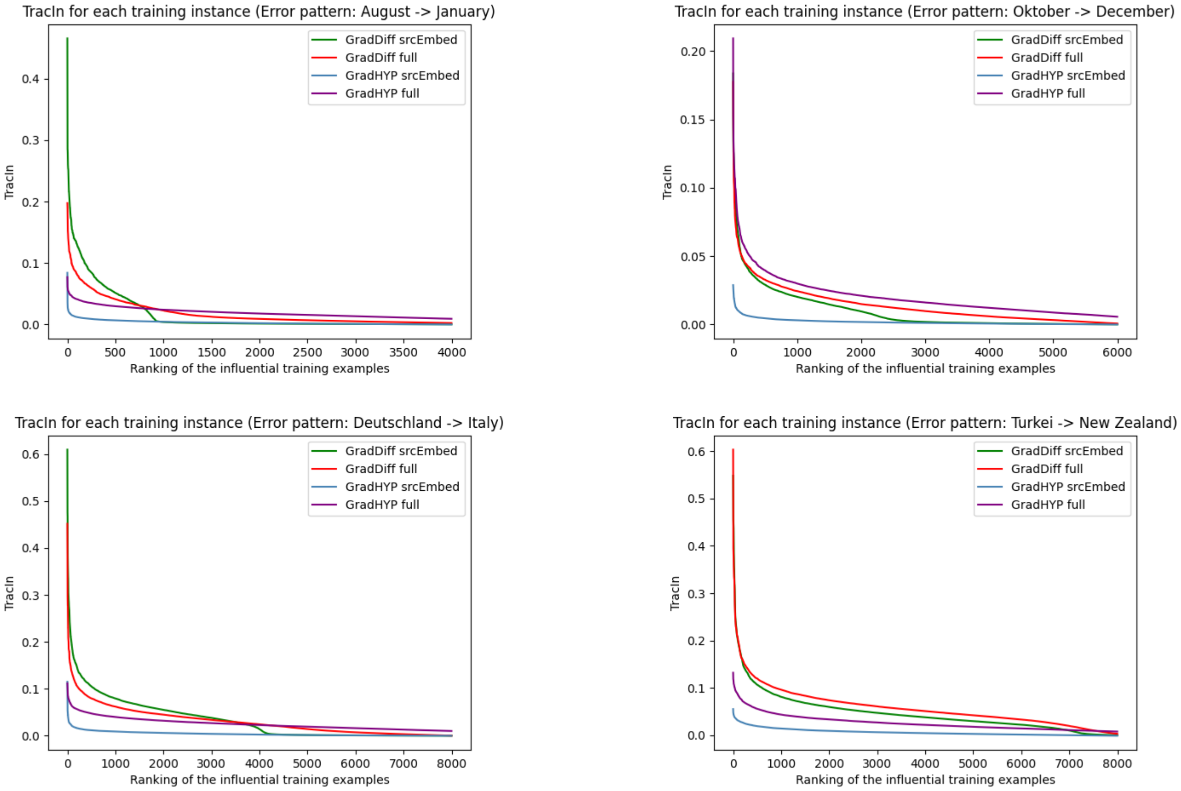


Figure A1: TracIn of the top-50% positively influential training examples. In each subfigure, we randomly select a probing example from each error pattern to compute its influence using gradient difference w.r.t. 1) source embedding (GradDiff srcEmbed), and 2) entire model (GradDiff full) as well as using vanilla-IF with source-hypothesis as input w.r.t. 1) source embedding (GradHYP srcEmbed), and 2) entire model (GradHYP full).

probing	1	Der Film läuft bei uns ab dem 25. August. The film will be filmed here on 25 January.
∇_{srcEmb}	1	Die Vereinbarung läuft am 31. Januar ab. This agreement formally expires on 31 January.
	2	Dieses Gesetz wurde im August unterzeichnet. It was signed in January.
	3	Die Vereinigten Staaten haben diese Garantie am 15. August 1971 aufgegeben. The United States abandoned that guarantee on 15 January 1971.
∇_{output}	1	Der Cardiff-Bericht erscheint Mitte Januar. The Cardiff report will be published in mid-January.
	2	Eine zweite Tagung ist für Januar 2004 vorgesehen. A second meeting will be held in January 2004.
	3	Ich hoffe, dass die Dynamik beibehalten und das Siebte Rahmenprogramm am 1. Januar 2007 auf den Weg gebracht wird. I hope that the momentum will be maintained and the Seventh Framework Programme will be launched on 1 January 2007.
probing	2	Auch in Deutschland finde eine "Hexenjagd" gegen Erdogan-Kritiker statt. A 'witch hunt' against Erdogan critics is also taking place in Italy.
∇_{srcEmb}	1	Deutschland ist dagegen. Italy is opposed to this.
	2	Dies wäre ein besseres Wirtschaftsmodell für Deutschland. This would be a better economic model for Italy.
	3	Deutschland und China können mehr tun als andere. Italy and China can do more than others.
∇_{output}	1	Eine weitere Lehre für Sarkozy aus Deutschland ist, dass ein aufgeklärter korporatistischer Staat unterstützender politischer Führung ebenso bedarf wie entgegenkommender Gewerkschaften. A further lesson for Sarkozy from Italy is that an enlightened corporate state needs supportive political leadership as well as accommodating trade unions.
	2	Insgesamt wurden fast 2 300 Tonnen möglicherweise kontaminiertes Futtermittelfett an 25 Futtermittelhersteller in Deutschland geliefert. A total of almost 2 300 tonnes of potentially contaminated feed fat was delivered to 25 feed manufacturers in Italy.
	3	Leider Gottes ist der Titel der heutigen Debatte Italien. Alas, the title of today's debate is Italy.

Table A2: Two probing examples with source-hypothesis as input and their top-3 positively influential training instances. ∇_{output} has a tendency to assign higher scores to sentence-pairs which target side has overlapped tokens but ignoring the similarity of the source side. For example, the pattern "*Januar -> January*" occurs more frequently in the ranking than "*August -> January*" in probing 1.

∇(Probing)	+/-	Precision ∇_{srcEmb}	$\nabla_{encoder}$	∇_{trgEmb}	∇_{output}	∇_{concat}	∇_{Full}
∇(HYP)	+	0.846	0.485	0.334	0.720	0.722	0.503
∇(REF)	-	0.876	0.432	0.303	0.794	0.805	0.481
∇(CorrHYP)	-	0.930	0.494	0.324	0.905	0.919	0.531
∇(HYP$_{\text{Mask}}$)	+	0.893	0.581	0.347	0.840	0.844	0.654
∇(HYP$_{\text{MaskExact}}$)	+	0.957	0.862	**0.474**	0.910	0.916	0.862
∇(CorrHYP$_{\text{MaskExact}}$)	-	0.989	0.903	0.467	**0.992**	**0.994**	0.924
∇(HYP) - ∇(REF)	+	0.930	0.523	0.321	0.856	0.855	0.584
∇(HYP) - ∇(CorrHYP)	+	**1.000**	**0.985**	0.458	0.971	0.980	**0.987**

(a) Retrieval performance for top-1% influential training examples

∇(Probing)	+/-	Precision ∇_{srcEmb}	$\nabla_{encoder}$	∇_{trgEmb}	∇_{output}	∇_{concat}	∇_{Full}
∇(HYP)	+	0.765	0.399	0.301	0.644	0.646	0.442
∇(REF)	-	0.799	0.382	0.297	0.693	0.700	0.437
∇(CorrHYP)	-	0.844	0.402	0.299	0.781	0.789	0.455
∇(HYP$_{\text{Mask}}$)	+	0.848	0.478	0.311	0.829	0.831	0.567
∇(HYP$_{\text{MaskExact}}$)	+	0.936	0.794	**0.380**	0.904	0.908	0.825
∇(CorrHYP$_{\text{MaskExact}}$)	-	0.962	0.821	0.372	**0.958**	**0.960**	0.875
∇(HYP) - ∇(REF)	+	0.855	0.442	0.307	0.764	0.765	0.515
∇(HYP) - ∇(CorrHYP)	+	**0.986**	**0.884**	0.371	0.935	0.939	**0.931**

(b) Retrieval performance for top-10% influential training examples

Table A3: Retrieval performance measured in (macro) averaged precision over all error patterns (extended version of Table 3). $\nabla(Probing)$ refers to the gradient with input 'source-Probing'. HYP, REF and CorrHYP stands for hypothesis, reference and corrected hypothesis respectively. "+" ("-") indicates that positively (negatively) influential training instances were retrieved. ∇_X indicates network components used in computing the gradient, ∇_{concat} indicates concatenation of ∇_{srcEmb}, ∇_{trgEmb} and ∇_{output}. We mark the best result per column in bold.

probing	1	Golfer Langer erhält die Sportpyramide Golfer Langer erhält die Sportpyramide
∇_{srcEmb}	1	Binnenmarktanzeiger Binnenmarktanzeiger
	2	Vollständige Liste der ausgewählten Aussteller: Vollständige Liste der ausgewählten Aussteller:
	3	Dimiter TZANTCHEV Ständiger Vertreter Dimiter TZANTCHEV Ständiger Vertreter
∇_{Full}	1	Erstellung einzelstaatlicher Aktionspläne für die Verhütung von Verletzungen durch die Mitgliedstaaten. Erstellung einzelstaatlicher Aktionspläne für die Verhütung von Verletzungen durch die Mitgliedstaaten.
	2	Für weitere Informationen wenden Sie sich bitte an die Dienststelle Außenbeziehungen Europäischer Rechnungshof Für weitere Informationen wenden Sie sich bitte an die Dienststelle Außenbeziehungen Europäischer Rechnungshof
	3	Dimiter TZANTCHEV Ständiger Vertreter Dimiter TZANTCHEV Ständiger Vertreter
probing	2	Die demokratische Bewerberin kündigt gar die größte Investition in neue Arbeitsplätze seit dem Zweiten Weltkrieg an. Die demokratische Bewerberin kündigt gar die größte Investition in neue Arbeitsplätze seit dem Zweiten Weltkrieg an.
∇_{srcEmb}	1	Die Krise hat die großen Unterschiede innerhalb der EU deutlich gemacht. Die Krise hat die großen Unterschiede innerhalb der EU deutlich gemacht.
	2	Die Regierungskonferenz ist nur eine Versammlung aller Regierungen. Die Regierungskonferenz ist nur eine Versammlung aller Regierungen.
	3	Die Entschließung wird uns dabei helfen, auf einer soliden Grundlage in die nächste Phase der Entwicklung einer Meeresstrategie einzutreten. Die Entschließung wird uns dabei helfen, auf einer soliden Grundlage in die nächste Phase der Entwicklung einer Meeresstrategie einzutreten.
∇_{Full}	1	Die Partei für Freiheit möchte dafür sorgen, dass die niederländische Öffentlichkeit nicht länger als Geldautomat Europas behandelt wird. Die Partei für Freiheit möchte dafür sorgen, dass die niederländische Öffentlichkeit nicht länger als Geldautomat Europas behandelt wird.
	2	Die russische Regierung hat geschätzt, dass ein Drittel aller Wasserleitungen dringend ersetzt werden muss. Die russische Regierung hat geschätzt, dass ein Drittel aller Wasserleitungen dringend ersetzt werden muss.
	3	Die internationale Gemeinschaft erkannte ihn einstimmig an. Die internationale Gemeinschaft erkannte ihn einstimmig an.

Table A4: Two probing examples with copied training instances as input and their top-3 positively influential training instances. Both ∇_{srcEmb} and ∇_{Full} can retrieve copied instances in the training subset given a probing instance of copied source sentence which is lexically different.

The AISP-SJTU Translation System for WMT 2022

Guangfeng Liu[1] **Qinpei Zhu**[1] **Xingyu Chen**[2] **Renjie Feng**[1] **Jianxin Ren**[1] **Renshou Wu**[1]
Qingliang Miao[1] **Rui Wang**[2] **Kai Yu**[1,2]
[1]AI Speech Co., Ltd., Suzhou, China
[2]Shanghai Jiao Tong University, Shanghai, China

Abstract

This paper describes AISP-SJTU's participation in WMT 2022 shared general MT task. In this shared task, we participated in four translation directions: English→Chinese, Chinese→English, English→Japanese and Japanese→English. Our systems are based on the Transformer architecture with several novel and effective variants, including network depth and internal structure. In our experiments, we employ data filtering, large-scale back-translation, knowledge distillation, forward-translation, iterative in-domain knowledge finetune and model ensemble. The constrained systems achieve 48.8, 29.7, 39.3 and 22.0 case-sensitive BLEU scores on EN→ZH, ZH→EN, EN→JA and JA→EN, respectively.

1 Introduction

We participate in the WMT 2022 shared general MT task, including English↔Chinese(EN↔ZH) and English↔Japanese(EN↔JA). All of our systems are built with constrained data sets.

For model architectures, we exploit several Transformer variants including transformer-DLCL (Wang et al., 2019), transformer-ODE (Bei Li, 2021), transformer-RPR (Shaw et al., 2018), transformer-Coda (Zheng et al., 2021).

In this year's translation tasks, we mainly employ data filtering (Zhou et al., 2021; Zeng et al., 2021), large-scale back-translation (Sennrich et al., 2015; Lample et al., 2017), knowledge distillation, forward-translation, in-domain knowledge finetune and model ensemble to improve the final model's performance.

For the synthetic data generation, we first exploit large-scale back-translation (Sennrich et al., 2015) method to leverage the target-side monolingual data and the knowledge distillation (Kim and Rush, 2016) to leverage the source-side of bilingual data. To use the source-side monolingual data, we explore forward-translation by ensemble models to get general domain synthetic data.Furthermore, several data augmentation methods are applied to improve the model robustness, including different token-level noise and different sampling methods.

We mainly use three training strategies in the training phase, including the warmup strategy (He et al., 2016) to adjust the learning rate in training, different sampling methods (Holtzman et al., 2019) and the Graduated Label Smoothing (Wang et al., 2020).

In the fine-tuning stage, the test set is clustered into seven categories, and then use the TFIDF-Ngram algorithm (Ramos et al., 2003) to search for similar bilingual and monolingual data in all data according to these seven domains. The monolingual data is then generated using forward translation to generate pseudo-data, and finally fine-tuned together with the searched bilingual data.

We pay more attention to the differences between different models in this year. We compute Self-BLEU (Zhu et al., 2018) from the translations of the models on the valid set to quantify the diversity among different models. To be precise, we use the translation of one model as the hypothesis and the translations of other models as references to calculate an average BLEU score. A lower Self-BLEU means this model is more different from other models.

For ensemble method in every category, the self-BLEU scores of the models are calculated to represent their differences from other models, and according to the self-BLEU scores of the model, the distribution weight when they perform ensemble is calculated through the Softmax-Temperature (Zhu et al., 2018; Cheng et al., 2017). Now seven domain ensemble models are obtained, then use the model for each domain to predict the test set of the corresponding domain separately.

This paper is structured as follows: Sec. 2 describes the novel model architectures. We introduce our system and training strategy in detail in Sec. 3. Experimental settings and results are shown in Sec. 4. We conduct analytical experiments in Sec. 5. Finally, we conclude our work in Sec. 6.

2 Model Architectures

2.1 Model Configurations

As the number of model parameters increases, the model's performance is better, so deeper and wider architectures are used in our system. However, the training of the deep model is unstable, and the loss is not easy to converge. Recent studies (Liu et al., 2020a; Huang et al., 2020) show that the unstable training problem of Post-Norm Transformer can be mitigated by modifying initialization of the network and the successfully converged Post-Norm models generally outperform Pre-Norm counterparts. We adopt the Admin initialization method (Liu et al., 2020b) in our training flows to stabilize the training of deep Post-Norm Transformer. Our experiments have shown that the Post-Norm model has a good diversity compared to the Pre-Norm model and slightly outperform the Pre-Norm model.

In our experiments, we use multiple model configurations with 24/30-layer encoders to build deeper models, and the decoder layers are all 6, and the hidden layer size of all models is 4096. Note that all model configurations above apply to the following variant models.

In addition, We use Transformer-ODE as the baseline model.

2.2 Transformer-RPR

According to the research of (Shaw et al., 2018), adding relative position representation to the self-attention mechanism is used to characterize the distance relationship of elements in the sequence, which can further improve the performance of the machine translation performance. So we incorporate relative position representation (RPR) into the self-attention mechanism on both the Transformer encoder and decoder side. Preliminary experiments demonstrate that only relative key information is enough, and we set the relative window size to 8.

2.3 Transformer-Coda

At the heart of the Transformer architecture is the Multi-Head Attention (MHA) mechanism which models pairwise interactions between the elements of the sequence. Despite its massive success, the current framework ignores interactions among different heads, leading to the problem that many of the heads are redundant in practice, which underutilizes the capacity of the model. To improve parameter efficiency, according to the research of (Zheng et al., 2021), we adopt cascaded head-colliding attention (CODA) which explicitly models the interactions between attention heads through a hierarchical variational distribution.

2.4 Transformer-DLCL

From the perspective of improving the residual network structure, we introduce the DLCL(Dynamic Linear Combination of Layers) method to solve the problem of gradient disappearance or explosion in deep model training. According to the research of (Wang et al., 2019), this DLCL method can effectively improve the performance of deep models.

2.5 Transformer-ODE

According to the research of (Bei Li, 2021), residual networks are an Euler discretization of solutions to Ordinary Differential Equations (ODE), and a residual block of layers in Transformer can be described as a higher-order solution to ODE. Inspired by this work, we adopt ODE to relieve the problem of gradient disappearance or explosion in deep model training.

3 System Overview

3.1 Data Filtering

For ZH-EN and JA-EN language pairs, the filtering rules are as follows:

- Filter out sentences which are longer than 120 words or contain a long word with over 40 characters.
- The word ratio between the source and the target sentence must not exceed 1:3 or 3:1.
- Filter out the sentences that have invalid Unicode characters or HTML tags.
- Filter out the duplicated sentence pairs.
- The number of punctuation difference between the source and the target sentence must not exceed 5.
- The number of digit difference between the source and the target sentence must not exceed 3.

* Filter out sentence pairs in which English sentence has Chinese or Japanese characters.

Besides these rules, several models are trained with constrained corpus for filtering corpus:

* Filter the bilingual corpus with semantic matching models.

* Filter the bilingual corpus with word align models (Dyer et al., 2013) .

* Filter out incomplete English sentences by a discriminative model.

* Filter out incomplete Japanese sentences by a discriminative model.

* Filter out classical Chinese and ancient poetry sentences by a discriminative model.

The monolingual corpus is also filtered with the above rules and models which are suitable for monolingual data. All the above rules and models are applied to synthetic parallel corpus as well.

3.2 Data Augmentation

In the field of NLP text classification, (Wei and Zou, 2019) proposed EDA technology, which can further improve the performance of the model. Inspired by this work, we introduce three operations of synonym replacement, random swap, and random deletion to generate new data. Here we call it **Aug**. Specifically, we choose 15% of sentence pairs to add noise and keep the remaining 85% of sentence pairs unchanged. For a chosen pair, we keep the target sentence unchanged, and perform the following three operations on the source sentence:

* 30% probability of synonym replacement.

* 50% probability of random swap.

* 20% probability of random deletion.

3.3 General Domain Synthetic Data Generation

In this section, we describe our methods for constructing general domain synthetic data. The general domain synthetic data is generated via large-scale back-translation, forward-translation and knowledge distillation to enhance the models' performance for all domains. In the following sections, we elaborate the above techniques in detail.

3.3.1 Back-Translation

Back-translation is the most commonly used data augmentation technique to make good use of the target side monolingual data in NMT (Hoang et al., 2018). Previous work (Edunov et al., 2018) has shown that Different generation strategies have different effects on the quality of generated pseudo-data. After these efforts, we employ the following three generation strategies.

* Sampling Top-K: At each time step, the model generates the probability that each word in the dictionary is likely to be the next word, which we randomly draw from a sample of k = 10 most likely candidates in this distribution. Afterwards, words are generated at the next time step based on the previously selected words.

* Sampling Top-P: Top-P Sampling (Nucleus sampling) is to preset a probability limit p-value, and then arrange all possible words from high to low according to the probability, and select words in turn. Stop when the cumulative probability of a word is greater than or equal to the p-value, and then sample from the already selected words to generate the next word. In our experiments, p is set to 0.9.

* Beam Search: Generate target translation by beam search with beam size 5.

Besides, we also use Tagged Back-Translation (Caswell et al., 2019) in En→Zh, Zh→En, En→Ja and Ja→En.

3.3.2 Forward-Translation

Forward translation refers to the generation of pseudo-data using source-side monolingual data (Sennrich et al., 2015). We use the ensemble model to generate high-quality forward translation data, which can greatly improve the robustness and performance of the model. Forward translation provides steady improvements on all four tracks we competed.

3.3.3 Knowledge Distillation

Knowledge Distillation (KD) has been proven to be a powerful technique for NMT (Kim and Rush, 2016; Wang et al., 2020) to transfer knowledge from the teacher model to student model. Specifically, we use an integrated teacher model to generate target-side pseudo-data from the source side

Domains	Zh	EN	JA
CLIENT	345	364	0
conversational	0	0	502
ecommerce	518	515	453
medicals	277	1454	0
news	505	1910	505
social	503	0	0
t1	0	279	191
t3	0	14343	305
voa	0	19	0

Table 1: The distribution of the blind test sentences in different domains.

of bilingual data. Likewise, Knowledge Distillation has steadily improved on all four tracks we participated in.

3.4 In-domain Finetune

Domain adaption (Luong and Manning, 2015) plays an important role in improving the translation performance. Different from the single domain (news) in previous years, the blind test of this year has shifted to a multi-domain. Firstly, we extract the domain information of every sentence from the "doc" tag in the XML files. The distribution of the blind test sentences in different domains is shown in Table 1. Secondly, we build 1-gram, 2-gram, 3-gram, 4-gram vocab for every domain and adopt the TF-IDF algorithm to extract fine-tuning data for each domain from the whole training set. Thirdly, we finetune the models for each domain using the corresponding domain data, 90% of which is used for training and 10% for validation. Finally, the models of each domain are ensembled and generate translation results of the test sentence in the corresponding domain.

3.5 Softmax-T Self-BLEU based Ensemble

After we get numerous fine-tuned models, we need to integrate them for better results. We improve on the traditional Self-BLEU method (Zhu et al., 2018). First, we calculate the Self-BLEU score of each model in each domain, and then obtain the weight score assigned to each model in each domain through the Softmax-Temperature (Zhu et al., 2018; Cheng et al., 2017). Finally, we use the models of the respective domains to integrate according to the assigned weight scores to generate data for the respective domains.

4 Experiments and Results

4.1 Settings

All our models are implemented based on fairseq 1.0.0. All the models are carried out on 8 NVIDIA V100 GPUs, each of which has 32 GB memory. We use the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$. We use an initial learning rate of 0.001 and use a warm-up strategy during the training phase. We use warm-up step = 4000. The max token is set to 3500 tokens per GPU and we set the "update-freq" parameter in Fairseq to 8. The value of the parameter Dropout is set to 0.3, and the value of Relu-Dropout is set to 0.1. We use the officially required sacreBleu to calculate all our models.

4.2 Dataset

The statistics of all training data is shown in Table 2. For each language pair, the bilingual data is the combination of all parallel data released by WMT22. For monolingual data, we select data from News Crawl, Common Crawl and Extended Common Crawl, and the amount of data after processing is shown in Table 2.

For generating pseudo-data, we use all source monolingual to generate forward translation data and all target monolingual to generate back-translation data. Finally we use the source side of bilingual data to generate knowledge distillation data. We use the methods described in Sec. 3.1 to filter bilingual and monolingual data.

4.3 Pre-processing and Post-processing

Before model training, we pre-process the training data uniformly and customize the processing according to the requirements of each model. Chinese sentences are segmented by Jieba [1], and English, we use Moses [2] for segmentation, and Japanese, we use Mecab [3]. Punctuation normalization is applied in Chinese, English and Japanese data. Truecasing is also applied for all the languages. For all the languages, we use byte pair encoding (BPE) with 40K operations to do subword segmentation (Sennrich et al., 2016).

For the post-processing, we apply de-tokenizing and de-trucaseing on the translation results with the scripts provided in Moses. And we use punctuation normalization for the Chinese and Japanese translations.

[1] https://github.com/fxsjy/jieba
[2] http://www.statmt.org/moses/
[3] https://github.com/taku910/mecab

Data	**En→Zh**	**Zh→En**	**En→Ja**	**Ja→En**
Bilingual Data	25M	25M	26M	26M
Source Mono Data	15M	15M	10M	10M
Target Mono Data	45M	45M	20M	20M

Table 2: Statistics of all training data

System	**En→Zh**	**Zh→En**	**En→Ja**	**Ja→En**
Baseline	45.0	31.0	38.5	23.2
+Back Translation	46.5	33.6	39.5	27.9
+Knowledge Distillation	47.0	34.7	-	-
+Forward Translation	47.4	35.0	40.2	28.2
+$OurIndomainFinetune$	47.5	-	-	**28.9**
+$NormalEnsemble$	48.0	35.7	40.3	28.3
+$OurEnsemble$	**48.1**	**35.9**	**40.4**	28.4

Table 3: Case-sensitive BLEU scores(%) on the four directions *newstest2020*. $OurEnsemble$ method outperform the $NormalEnsemble$. $OurIndomainFinetune$ prove to be effective through validation in the news domain. The final submitted system is a $OurEnsemble$ of all models which are finetuned in each domain using $OurIndomainFinetune$.

BASELINE-MODEL	**En→Zh**	**Zh→En**	**En→Ja**	**Ja→En**
Transformer	44.0	30.5	37.7	22.3
Transformer#	44.3	30.7	37.9	22.6
Transformer-RPR#	44.6	31.0	38.0	22.8
Transformer-Coda#	45.2	31.1	38.1	22.8
Transformer-DLCL#	45.3	31.3	38.2	23.0
Transformer-ODE#	45.0	31.0	38.5	23.2

Table 4: Case-sensitive BLEU scores (%) on the four translation directions *newstest2020* for different architecture in the **baseline** stage. The model with # indicates that the initialized strategy is ADMIN.

BASELINE-MODEL	**En→Zh**	**Zh→En**	**En→Ja**	**Ja→En**
Transformer#	44.3	30.7	37.9	22.6
Transformer-SourceAug#	44.8	31.1	38.2	23.0
Transformer-TargetAug#	44.6	30.6	37.9	22.5
Transformer-BothAug#	44.2	30.5	37.7	22.4

Table 5: Case-sensitive BLEU scores (%) on the four translation directions *newstest2020* for different **data augmentation methods** in the **baseline** stage.

4.4 English→Chinese

The results of En→Zh on *newstest2020* are shown in Table 3. For the En→Zh task, there is a significant improvement in the valid set after adopting our data filtering method. Our baseline score is 45.0. After applying large-scale Back-Translation, we obtain +1.5 BLEU score on the baseline. We further gain +0.5 BLEU score after applying knowledge distillation and +0.4 BLEU from forward-translation.

In preliminary experiments, we select all models distilled from knowledge as our ensemble combinations obtaining +0.6 BLEU score. On top of that, We tried various combinations but couldn't get better results. After using our proposed ensemble strategy, the BLEU score continue to improve by 0.1, which saves a lot of manpower to select models.

4.5 Chinese→English

The Zh→En task follows the same training procedure as En→Zh. As shown in Table 3, we can observe that Back-Translation can improve 2.6 BLEU from baseline. After this, knowledge distillation brings a big improvement, which can increase the BLEU scores from 33.6 to 34.7. Forward translation further boosts the BLEU score to 35.0. Likewise, our ensemble strategy saves a lot of manpower while delivering a small BLEU boost, from 35.7 to 35.9.

4.6 English→Japanese

The results of En→Ja on *newstest2020* are shown in Table 3. The bilingual training data is 31M in total, and we filter it down to 26M sentence pairs through the filtering rules and models described earlier. Because *newstest2020* has detailed results of each step as a reference, we regard the *newstest2021* as the valid set and the *newstest2020* as the test set during training. The 26 million bilingual training data brings the baseline model to 38.5 BLEU score on *newstest2020*.

For the back translation, our training data consists of three parts: 1) 26 million bilingual target data, 2) Japanese monolingual data, 3) Bilingual augmented data. In addition to the 26 million bilingual target sentences, we sample 20 million Japanese monolingual data from the combination of News Crawl and Common Crawl. Then we used the JA-EN ensemble model to generate the hypotheses as the pseudo data set via the Top-k, Top-p and beam search strategy. We randomly extract 2 million from the bilingual data, and add noise to the source sentences as described in Sec 3.2. We improve BLEU by 1.0 with the synthetic back translation training data.

And then, we merge knowledge distillation and forward translation together. We extract 26 million bilingual source sentences and 10 million source monolingual data, and generate pseudo data using the ensemble model of the back translation models. We also use 2 million noised data like used in back translation. We improve the BLEU score from 39.5 to 40.2.

In the ensemble stage, we observe that both of the normal ensemble and our ensemble strategy have only a very slight improvement.

4.7 Japanese→English

The Ja→En task follows the same training procedure as En→Ja. From Table 3, we can observe that back translation can improve the BLEU score from 23.2 to 27.9. The knowledge distillation and forward translation further improve 0.3 BLEU score. In this task, we verify the effectiveness of our in-domain fine tuning method in the News domain. It is worth mentioning that out in-domain fine tuning method brings 0.7 BLEU after forward-translation. For the comparability of the experiment, we still ensemble models which are on the base of forward-translation. We observe that both ensemble methods make results worse.

5 Analysis

To verify the effectiveness of our approach, we conduct analytical experiments on model variants, data augmentation methods, and ensemble strategies in this section.

5.1 Effects of Model Architecture

We conduct several experiments to validate the effectiveness of Transformer (Vaswani et al., 2017) variants we used in the baseline stage and list results in Table 4. Here we take the En→Zh and En→Ja models as examples to conduct the experiments. The results in the Zh→En direction are similar to En→Zh, and the results for the Ja→En direction are similar to En→Ja.

As shown in Table 4, Transformer-DLCL achieves the best performance in En→Zh direction, and Transformer-ODE achieves the best performance in En→Ja direction. For Admin (Liu et al., 2020b) initialization, Transformer#'s BLEU is 0.2 higher than Transformer in En→Zh and En→Ja directions, so this verifies the effectiveness of Admin initialization in deep models.

5.2 Effects of Data Augmentation

For data augmentation, we conduct several experiments based on the Transformer# baseline model in four directions. Specifically, we adopt three methods detailed in Section 3.2:

* **SourceAug** Aug on the source text of the sentence pair.

* **TargetAug** Aug on the target text of the sentence pair.

* **BothAug** Aug on the source text and target text of the sentence pair.

The experimental results are shown in Table 5. Taking En→Zh direction as an example, the SourceAug achieves a BLEU score of 44.8, TargetAug achieves a BLEU score of 44.6, and BothAug achieves 44.2. Results in other directions show the same trend. Therefore, we operate on the source text of sentence pairs in the data augmentation process.

6 Conclusion

This paper summarizes the results of the shared general MT task in the WMT 2022 produced by the AISP-SJTU team. In this shared task, we participated in four translation directions: English→Chinese, Chinese→English, English→Japanese and Japanese→English. We investigate various novel Transformer based architectures to build MT systems. Our systems are also built on several popular data augmentation methods such as back-translation, knowledge distillation, forward-translation and in-domain finetune. In the future, we hope to explore more efficient model architectures and data augmentation techniques in MT systems. We hope that our practice can facilitate research work and industrial applications.

References

Tao Zhou Shuhan Zhou Xin Zeng Tong Xiao Jingbo Zhu Bei Li, Quan Du. 2021. Ode transformer: An ordinary differential equation-inspired model for neural machine translation. *arXiv preprint arXiv:2104.02308*.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. *arXiv preprint arXiv:1906.06442*.

Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. 2017. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv:1710.09282*.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *NAACL*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd workshop on neural machine translation and generation*, pages 18–24.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Xiao Shi Huang, Felipe Perez, Jimmy Ba, and Maksims Volkovs. 2020. Improving transformer optimization through better initialization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4475–4483. PMLR.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. 2020a. Understanding the difficulty of training transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*.

Xiaodong Liu, Kevin Duh, Liyuan Liu, and Jianfeng Gao. 2020b. Very deep transformers for neural machine translation. *arXiv preprint arXiv:2008.07772*.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign*.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. 2019. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*.

Shuo Wang, Zhaopeng Tu, Shuming Shi, and Yang Liu. 2020. On the inference calibration of neural machine translation. *arXiv preprint arXiv:2005.00963*.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Xianfeng Zeng, Yijin Liu, Ernan Li, Qiu Ran, Fandong Meng, Peng Li, Jinan Xu, and Jie Zhou. 2021. Wechat neural machine translation systems for wmt21. *arXiv preprint arXiv:2108.02401*.

Lin Zheng, Zhiyong Wu, and Lingpeng Kong. 2021. Cascaded head-colliding attention. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 536–549, Online. Association for Computational Linguistics.

Shuhan Zhou, Tao Zhou, Binghao Wei, Yingfeng Luo, Yongyu Mu, Zefan Zhou, Chenglong Wang, Xuanjun Zhou, Chuanhao Lv, Yi Jing, et al. 2021. The niutrans machine translation systems for wmt21. *arXiv preprint arXiv:2109.10485*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.

NT5 at WMT 2022 General Translation Task

*Makoto Morishita[♠], *Keito Kudo[◇], *Yui Oka[♠], *Katsuki Chousa[♠],
†Shun Kiyono[♡], †Sho Takase[♣], Jun Suzuki[◇♡]
[♠]NTT Communication Science Laboratories [◇]Tohoku University
[♡]RIKEN Center for Advanced Intelligence Project [♣]Tokyo Institute of Technology

Abstract

This paper describes the NTT-Tohoku-TokyoTech-RIKEN (NT5) team's submission system for the WMT'22 general translation task. This year, we focused on the English-to-Japanese and Japanese-to-English translation tracks. Our submission system consists of an ensemble of Transformer models with several extensions. We also applied data augmentation and selection techniques to obtain potentially effective training data for training individual Transformer models in the pre-training and fine-tuning scheme. Additionally, we report our trial of incorporating a reranking module and the reevaluated results of several techniques that have been recently developed and published.

1 Introduction

This paper describes an overview of our submission systems for participating in the WMT 2022 general machine translation tasks. Our team, named NT5, is comprised of individuals from four organizations: NTT, Tohoku University, Tokyo Institute of Technology, and RIKEN. This year, we focused on bi-directional translation in a single language pair: English-to-Japanese (En→Ja) and Japanese-to-English (Ja→En) translation tracks.

Our submission system consists of an ensemble of Transformer models (Vaswani et al., 2017) with several recent extensions. We also applied data augmentation and selection techniques to obtain potentially effective training data for training individual Transformer models in the pre-training/fine-tuning scheme. The models were first trained with a large but possibly noisy parallel corpus for pre-training and then with a small but clean parallel corpus for fine-tuning. Additionally, we report our trial of incorporating a reranking module that rescores the n-best lists based on source-to-target, target-to-source, and masked language models.

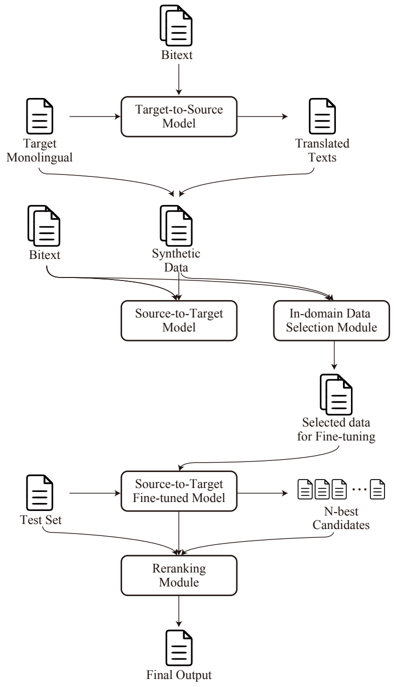


Figure 1: System overview

The following section briefly provides an overview of our entire system and each module in more depth.

2 System Overview

Figure 1 shows an overview of our system. Our submissions are for the **constrained track**, which only uses parallel and monolingual data that are provided by the WMT shared-task organizers.

*: Equal contributions. Morishita trained the initial translation model and created a synthetic corpus. Morishita and Kudo tuned the model's hyper-parameters. Oka implemented the relative positional embeddings. Chousa worked on the data filtering and the reranking module. Kiyono and Takase implemented the B2T connections and helped with the experiments. Morishita and Oka implemented the in-domain data selection methods for fine-tuning. Morishita, Kudo, and Suzuki developed an effective fine-tuning strategy. Morishita, Kudo, and Suzuki trained the main translation models. Suzuki organized the team. Everyone contributed to writing this paper.

†: Their current affiliation is LINE Corporation.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 318 - 325
December 7-8, 2022 ©2022 Association for Computational Linguistics

We selected Transformer models (Vaswani et al., 2017) as our base translation model and chose a two-step training strategy: pre-training and fine-tuning schemes. We first constructed datasets for the pre-training and fine-tuning.

The pre-training dataset must be as large as possible, even if the data are noisy (Bansal et al., 2022). We first trained the Transformer models using only the provided bitext datasets for both translation directions: En→Ja and Ja→En. We refer to these first trained models as **initial models**. We then generated synthetic datasets for both directions through back-translation (Sennrich et al., 2016), i.e., translating target-side monolingual data using the initial model in the reverse translation direction.

The fine-tuning dataset must be as clean as possible, even if it is relatively small. Indeed, in our previous year's submission (Kiyono et al., 2020), we adapted the models to a news domain in the fine-tuning phase and drastically improved the translation quality. However, this year's task focused on a general domain, i.e., a test set that consisted of sentences from multiple domains. Thus, adaptation by fine-tuning is much more challenging. We tested and combined data selection methods based on sentence embeddings and language models for obtaining fine-tuning data. This process can be viewed as selecting domain adaptation data.

By using these datasets, we pre-trained the Transformers with pre-training configurations and fine-tuned the pre-trained models with the fine-tuning configurations described in Table 2. Finally, we conducted an ensemble of fine-tuned models. A notable characteristic of our system is that we combined the Transformer models with heterogeneous model configurations for the ensembling. Each model configuration primarily differs in its depth and width. Moreover, we applied recent advances in the extensions of Transformer models, such as bottom-to-top connection (Takase et al., 2022) and relative position embedding (Shaw et al., 2018).

Our system also uses a reranking module. We generated the ten best translation lists as reranking candidates using an ensemble of Transformer models. Then we selected the best translations based on the weighted sum of the likelihoods obtained from the source-to-target and target-to-source translation models and the masked language models.

Corpus	w/o Filtering	w/Filtering
JParaCrawl v3.0	25.7 M	25.0 M
WikiMatrix	3.89 M	3.64 M
JESC	2.80 M	2.57 M
Wiki Titles v3	757 K	327 K
KFTT	440 K	371 K
TED Talks	242 K	224 K
NewsCommentary v16	1.9 K	1.8 K

Table 1: Number of sentence pairs in bitext corpus

3 Dataset Construction

3.1 Provided Data

Bitext Corpus We used all the provided bitext corpora: JParaCrawl v3.0, News Commentary v16, Wiki Titles v3, WikiMatrix, Japanese-English Subtitle Corpus (JESC), The Kyoto Free Translation Task (KFTT) Corpus, and TED Talks. We filtered out the potentially noisy pairs using the straightforward parallel corpus filtering methods, as described in Section 3.2. Table 1 shows the size of each dataset without/with filtering.

Monolingual Corpus We also used the following provided monolingual data: News Crawl, News Commentary, and Common Crawl. We back-translated the monolingual sentences with a target-to-source model trained only with the provided parallel data, as described in Section 3.2, and used them as synthetic data (Sennrich et al., 2016).

3.2 Building Pre-training Data

Synthetic Data Construction To augment the training data, we constructed synthetic data by applying the initial translation model trained with bitext to the monolingual data. As a preprocessing step, we truecased[1] both the bitext and monolingual data. We then tokenized the data into subwords using the `Sentencepiece` tool (Kudo and Richardson, 2018) with the unigram language model option. We set the vocabulary size to 32,000 for the initial translation model, which is used for creating synthetic data. For the final submission model, we increased the vocabulary size to 64,000. Our hypothesis argues that a bigger vocabulary is crucial for completely exploiting large synthetic data. In fact, this 64,000-vocabulary model outperformed the 32,000-vocabulary model in our preliminary experiment.

[1] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/recaser/truecase.perl

Initial Translation Model	
Subword Size	32,000
Architecture	Transformer (big) with FFN size of 4,096
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$)
Learning Rate Schedule	Inverse square root decay
Warmup Steps	4,000
Max Learning Rate	0.001
Dropout	0.3
Gradient Clip	1.0
Batch Size	1,280,000 tokens
Number of Updates	50,000 steps
Averaging	Save a checkpoint every 200 steps and average the last eight
Implementation	`fairseq` (Ott et al., 2019)
Pre-training Configuration	
Subword Size	64,000
Architecture	(See Table 4)
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$)
Learning Rate Schedule	Inverse square root decay
Warmup Steps	4,000
Max Learning Rate	0.001
Dropout	0.1
Gradient Clip	0.1
Batch Size	1,024,000 tokens
Maximum Number of Updates	100,000 steps
Averaging	Save a checkpoint every 2,000 steps and average the last ten
Implementation	`fairseq` (Ott et al., 2019)
Fine-tuning Configuration	
Subword Size	Identical to Pre-training Configuration
Architecture	(See Table 4)
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$)
Learning Rate Schedule	Fixed
Warmup Steps	N/A
Max Learning Rate	0.00001
Dropout	0.2
Gradient Clip	1.0
Batch Size	14,400 tokens
Number of Updates	Tuned for each model (See Secsion 4.3)
Averaging	Save a checkpoint every ten steps and average the last ten
Implementation	`fairseq` (Ott et al., 2019)

Table 2: List of hyper-parameters: We used initial translation model for creating synthetic data, pre-training configuration to construct pre-training models described in Section 4.2, and fine-tuning configuration to construct models for submission. We used several different model configurations for ensembling. See Table 4 for more details.

	#sent. pairs	#subwords (JA)	#subwords (EN)
En→Ja	579 M	11.6 B	13.1 B
Ja→En	724 M	15.5 B	16.7 B

Table 3: Statistics of synthetic data used for pre-training

As the initial translation data, we trained the Transformer-big model defined in the original Transformer paper (Vaswani et al., 2017) for both language directions (English-to-Japanese and Japanese-to-English translations) only with the provided bitext data. The detailed hyper-parameters are described in the initial translation model section of Table 2. Finally, we respectively translated 1.4B and 1.2B monolingual sentences for English and Japanese.

Data Cleaning For both the provided bitext and synthetic data, we carried out cleaning based on a combination of sentence embeddings and handcrafted rules.

For both the bitext and synthetic data, we removed the too-long sentences whose length exceeded 500 characters. We also removed the sentences that were identified as not being written in English or Japanese with the `langid`[2] toolkit.

For the synthetic data, we further applied a sentence embedding-based filtering approach. We took advantage of LaBSE (Feng et al., 2022) to embed the Japanese and English sentences into the same embedding space. We then scored and ranked the parallel sentence pairs based on the cosine similarity of their sentence embeddings. Subsequently, we filtered out the following items from the synthetic data:

- duplicated sentence pairs
- sentences over 150 words[3] or single words with over 40 characters
- sentences whose ratio between word and character count is greater than 12
- sentences that contain invalid Unicode characters
- sentence pairs whose source/target word ratio exceeds 4
- sentence pairs whose source/target length ratio exceeds 6
- sentence pairs whose source and target sentences are identical
- sentence pairs whose cosine similarity is greater than 0.96[4]

[2] `https://github.com/saffsd/langid.py`

[3] We tokenized the Japanese sentences by MeCab (Kudo, 2006) with the IPA dictionary. Note that this tokenization is for cleaning purpose only.

[4] We found that sentence pairs with high cosine similarities might be noisy; for example, the source and target sentences

Finally, we respectively selected approximately the top 579 M and 724 M sentences from the translated 1.2B and 1.4B monolingual sentences as the synthetic data of En→Ja and Ja→En in the rank orders. Table 3 shows the statistics of synthetic data used for our pre-training.

3.3 Building Fine-tuning Data

As for the fine-tuning data, we prepared two types of data: *news* and *general*. The news data consist of the dev and test sets of the WMT'20 news translation task, which has 3,991 sentences. General data were created by selecting parallel sentences in the target domain. We used the n-gram language model-based method proposed by Moore and Lewis (2010) and selected the top 20,000 scored sentences from the synthetic corpus. We also used sentence embeddings to select the general domain data. We used an unsupervised SimSCE (Gao et al., 2021) as the English sentence embedding and SentenceTransformers (Reimers and Gurevych, 2019) as the Japanese sentence embedding[5]. We searched for the nearest 4,000 sentences to the target domain using `faiss` (Johnson et al., 2019) and combined the sentences selected by both the language model-based and sentence embeddings. As a result, our general domain data contained 24,000 sentences.

4 Primary Translation Module

4.1 Model Configuration

We trained several Transformer models for the model ensembling in the decoding phase. We independently trained models with different sizes due to the restrictions on computational resources at hand. We pre-trained and fine-tuned each model with the configurations shown in Table 2. The details of the model configurations are summarized in Table 4.

Our configuration has three notable characteristics: a bottom-to-top (B2T) connection (Takase et al., 2022), relative position embedding, and a larger batch size.

B2T Connection Transformer architectures can be categorized into two types based on the position of the layer normalizations: Post-LN and Pre-LN. Previous studies (Xiong et al., 2020; Liu et al., 2020; Takase et al., 2022) indicated that training a deep Post-LN Transformer[6] is unstable due to the vanishing gradient problem. However, Takase et al. (2022) argued that Post-LN Transformers outperform Pre-LN Transformers if their trainings are successful. Thus, we want to exploit the advantage of Post-LN Transformers. In addition, we want to make our Transformers as deep (and wide) as possible to make a full use of large synthetic data.

Several studies proposed techniques that stabilize the trainings of Post-LN Transformers while retaining their performance advantages (Liu et al., 2020; Takase et al., 2022). In this study, we used the B2T connection proposed by Takase et al. (2022), which has an additional residual connection from an input to an output in each layer. The B2T connection is easy to implement and can be incorporated with a tiny amount of extra computational cost.

Relative Position Embedding A Transformer model was originally equipped with Absolute Position Embedding (APE) (Gehring et al., 2017) for position representation. However, several recent studies (Raffel et al., 2020; Narang et al., 2021) report that Relative Position Embedding (RPE) (Shaw et al., 2018) outperforms APE, especially for sentences whose lengths are unseen during the training (Kiyono et al., 2021). Thus, for the Transformer encoder, we replaced APE with RPE. Following Shaw et al. (2018), we set clipping distance k to 16.

Larger Batch Size Ott et al. (2019) demonstrated that a large batch size improves performance. The recent development of large language models also indicates this tendency (Hoffmann et al., 2022). Given this knowledge, we followed the setting of T5 (Raffel et al., 2020) and selected a token batch size of approximately 1M. Note that this is much larger than the batch size used by Ott et al. (2019).

4.2 Pre-training

We trained each model described in Table 4 with the filtered bitext and synthetic data described in Section 3.2. We set the maximum number of updates to 100,000 and used early stopping based on the validation set performance. In this phase, we used the Pre-training configuration of Table 2. Since the synthetic data is extremely larger than the

are sometimes identical. Thus we removed them from the training data.

[5] We used `stsb-xlm-r-multilingual`.

[6] When we used the dimension sizes described in Table 4, the trainings of nine or more layers of Post-LN Transformers diverged.

Configuration	#Models	#Params.	Encoder				Decoder			
			Layer	d_{model}	d_{ffn}	Attention Heads	Layer	d_{model}	d_{ffn}	Attention Heads
`NTT-Base`	2	547M	9	1024	8192	16	9	1024	8192	16
`ABCI-Base`	2	622M	9	1024	16384	16	9	1024	4096	16
`ABCI-EncBig`	1	2.0B	12	1024	65536	16	9	1024	8192	16
`ABCI-EncDeep`	1	736M	18	1024	8192	16	9	1024	8192	16
`Failab-EncBig`	1	1.7B	9	1024	61440	16	9	1024	16384	16
`Failab-DecBig`	1	1.7B	9	1024	16384	16	9	1024	61440	16

Table 4: List of model configurations used in final system: d_{model} and d_{ffn} respectively denote sizes of embedding and feedforward layers. In En→Ja, `Failab-EncBig` and `Failab-DecBig` did not fit in the GPU memory. Therefore, we set d_{ffn} to 58368 instead of 61440, which is the largest value that successfully worked.

bitext, we upsampled the bitext until it reaches to a 1:1 ratio to the synthetic data. In addition, we used the tagged back-translation technique (Caswell et al., 2019). In detail, we attached a special token ⟨`BT`⟩ to the beginning of source sentences in synthetic data.

To improve the performance, we tried using several perturbation methods described in Takase and Kiyono (2021) in this training phase. However, they did not positively affect the performance. Since the number of sentences in our training data is far greater than in their study, regularization by perturbations might be ineffective.

4.3 Fine-tuning

We fine-tuned the pre-trained translation models with the fine-tuning dataset described in Section 3.3 and used the configurations described in Table 2. We set the maximum number of updates to 600, and used early-stopping according to the performance on the test data of WMT'21 (wmt21test).

4.4 Ensemble

We ensembled the fine-tuned models described in Table 4[7]. How we ensembled the Transformer models trained in different model configurations is another unique characteristic of our system compared with the standard configurations used in the WMT submission systems.

5 Post-processing

5.1 Reranking

We tried to apply a reranking method to select the most likely candidate from a set of candidates and input. We scored the candidate with several models and unified these scores with Minimum Error Rate Training (MERT) (Och, 2003), which is often used in Statistical Machine Translation (SMT).

Suppose we have set of candidate output sentences C_i for each source sentence s_i, where $i \in \{1, \dots, I\}$. In our case, we generated n-best candidates using the submission model with the beam-search algorithm.

Hereafter, $P_j(s_i, e) \in [0, 1]$ denotes candidate score $e \in C_i$ for i-th input s_i from the j-th model, where $j \in \{1, \dots, J\}$, and $\boldsymbol{w} = (w_1, \dots, w_j)$ denotes the vector representation of the model weights. Given weights $\boldsymbol{w}$, the most likely candidate $\hat{e}_i^{\boldsymbol{w}}$ from C_i is obtained by maximizing the weighted sum of P_j:

$$\hat{e}_i^{\boldsymbol{w}} = \underset{e \in C_i}{\operatorname{argmax}} \left\{ \sum_{j=1}^{J} w_j P_j(s_i, e) \right\}. \quad (1)$$

Finally, we explored $\hat{\boldsymbol{w}}$ for the parameter estimation of $\boldsymbol{w}$ by solving the following optimization problem:

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w} \in [0,1]^J}{\operatorname{argmax}} \left\{ \texttt{corpus\_bleu}(\hat{\mathcal{E}}^{\boldsymbol{w}}) \right\}, \quad (2)$$

where $\hat{\mathcal{E}}^{\boldsymbol{w}} = \left(\hat{e}^i_{\boldsymbol{w}}\right)_{i=1}^{I}$.

For the candidate's score, we used the following models to compute $P_j(s_i, e)$.

L2R Forward and Backward Translation Models The left-to-right (L2R) forward and backward translation models are identical as those used for the candidate generation of En→Ja and Ja→En. For each direction, we trained two models with two different training data; these four models computed the score by force-decoding a candidate from their input.

R2L Forward and Backward Translation Models The right-to-left (R2L) forward and backward translation models generate a translation in reverse

[7] We trained two models with both the `NTT-base` and `ABCI-base` configurations with different random seeds.

ID	Model	En→Ja			Ja→En		
		wmt20dev	wmt21test	wmt22test	wmt20dev	wmt21test	wmt22test
(a)	`NTT-Base` (bitext only)	22.5	-	-	22.7	-	-
(b)	`NTT-Base` (Seed#1)	23.9	25.6	-	24.1	21.5	-
(c)	`NTT-Base` (Seed#2)	23.7	25.5	-	24.0	21.5	-
(d)	`ABCI-Base` (Seed#1)	25.4	27.6	-	**25.5**	**23.4**	-
(e)	`ABCI-Base` (Seed#2)	**26.0**	**28.3**	-	**25.5**	23.2	-
(f)	`ABCI-EncBig`	24.7	26.7	-	**25.5**	22.6	-
(g)	`ABCI-EncDeep`	24.8	26.5	-	25.3	22.8	-
(h)	`Failab-EncBig`	24.6	26.3	-	23.7	20.4	-
(i)	`Failab-DecBig`	23.5	25.4	-	23.0	20.9	-
(j)	(b), finetuned on news	-	28.6	26.3	-	25.6	24.9
	(c), finetuned on news	-	29.0	26.2	-	26.0	25.1
	(d), finetuned on news	-	28.9	26.6	-	25.8	25.4
	(e), finetuned on news	-	28.5	26.6	-	25.8	25.0
	(f), finetuned on news	-	**29.4**	26.5	-	**27.0**	25.5
	(g), finetuned on news	-	28.8	**26.7**	-	26.4	**25.6**
	(h), finetuned on news	-	29.2	**26.7**	-	25.6	25.2
	(i), finetuned on news	-	28.6	26.4	-	25.9	25.1
(k)	(b), finetuned on news+general	-	27.8	25.4	-	24.9	23.9
	(c), finetuned on news+general	-	28.0	24.8	-	24.5	23.8
	(d), finetuned on news+general	-	28.4	25.3	-	25.0	24.7
	(e), finetuned on news+general	-	28.0	25.3	-	24.9	24.6
	(f), finetuned on news+general	-	28.0	25.0	-	25.8	24.8
	(g), finetuned on news+general	-	28.5	25.6	-	25.3	24.5
	(h), finetuned on news+general	-	28.6	25.1	-	25.2	24.2
	(i), finetuned on news+general	-	27.9	24.7	-	25.1	23.9
(l)	Ensemble of (j)	-	**30.6**	**27.6**	-	27.8	**26.6**
(m)	Ensemble of (k)	-	29.6	25.8	-	26.8	25.4
(n)	Ensemble of (l) and (m)	-	**30.6**	27.2	-	**27.9**	**26.6**
(o)	(n) + reranking	-	-	25.7	-	-	25.0

Table 5: Performance comparison of models trained for submission: Models (b)-(i) are pre-trained models (details in Section 4.2). Models (j)-(o) do not contain wmt20dev result because the dataset is in news fine-tuning dataset. We chose model (l) for the final submission. Note that the wmt22test results were computed as the post-evaluation after the wmt22 test data was released.

word order. We trained the model of both directions with all the provided bitext datasets and computed the scores with the same procedure as was used for the L2R models.

Masked Language Models We also used the masked language models to compute the likelihood of the decoded target sentences. Specifically, we used the pre-trained models of DeBERTa (He et al., 2021)[8] for English and RoBERTa (Liu et al., 2019)[9] for Japanese. To score the candidate, we adopted *pseudo-log-likelihood scores* (PLLs), computed by masking tokens one by one, as proposed by Salazar et al. (2020) Finally, we normalized the PLLs by dividing them by token length.

5.2 Rule-base Formatting

We also applied language-specific post-processing.

Ja→En We detokenized and detruecased the sentences and removed all the unknown tokens from the outputs. Since some placeholders were tokenized into two or more tokens, we fixed them to a single token.

En→Ja We removed the spaces in the English proper nouns of two characters or fewer, the spaces before and after such special symbols as "/", "-" or "#PRS/ORG#". We replaced English style commas "," and periods "." with the Japanese styles: "，" and "。".

6 Results

Table 5 shows the performance of both the intermediate models and the final model for our submission. Our result highlights the effectiveness of the

[8] `https://huggingface.co/microsoft/deberta-v2-xxlarge`

[9] `https://huggingface.co/nlp-waseda/roberta-large-japanese-seq512`

techniques incorporated in our system.

Effectivenss of Fine-tuning Data We expected model (m), which was fine-tuned on the general domain, to achieve the best result. However, the model (l), which was fine-tuned on the news domain, achieved the higher BLEU score on both wmt21test and wmt22test. We suspect this is because the data used for the news fine-tuning are cleaner than those of the general domain. Since the fine-tuning data for the news domain consists of the previous years' dev/test sets that were translated by professionals, the news domain data are clean while the general domain data were chosen mainly from synthetic data. We will analyze the relationship between the translation accuracy and the cleanliness of the fine-tuned data in the future.

Negative Result on Reranking In Table 5, the performance of model (n) and (o) demonstrate that the reranking technique (Section 5.1) did not improve the performance over the ensemble models on wmt22test. We suspect that this performance degradation comes from the domain difference between the datasets used for MERT and the evaluation. For MERT, We used wmt21test, whose domain is news, to optimize the model weights; however, this year's test set, wmt22test, contains sentences from multiple domains. Thus, we chose model (l), which is the model without reranking, for our final submission.

7 Conclusion

We described the submission of our joint team (NTT, Tohoku, TokyoTech, and RIKEN) to the WMT'22 general translation task. We participated in the En↔Ja translation. Our system mainly consists of an ensemble of Transformer models with several recent extensions. We also applied data augmentation and selection techniques to train individual Transformer models in our pre-training/fine-tuning training scheme.

Acknowledgments

We thank two anonymous reviewers who provided valuable feedback. This work was mainly done under the NTT-Tohoku University collaborative research agreement. The work of Sho Takase was partly supported by JSPS KAKENHI Grant Number JP21K17800 and JST ACT-X Grant Number JPMJAX200I. The work of Jun Suzuki was partly supported by JST Moonshot R&D Grant Number JPMJMS2011 (fundamental research).

References

Yamini Bansal, Behrooz Ghorbani, Ankush Garg, Biao Zhang, Colin Cherry, Behnam Neyshabur, and Orhan Firat. 2022. Data scaling laws in NMT: The effect of noise and architecture. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 1466–1482.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged Back-Translation. In *Proceedings of the Fourth Conference on Machine Translation (WMT)*, pages 53–63.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 878–891.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6894–6910.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *Proceedings of 9th International Conference on Learning Representations (ICLR)*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Shun Kiyono, Takumi Ito, Ryuto Konno, Makoto Morishita, and Jun Suzuki. 2020. Tohoku-AIP-NTT at WMT 2020 news translation task. In *Proceedings of the Fifth Conference on Machine Translation (WMT)*, pages 145–155.

Shun Kiyono, Sosuke Kobayashi, Jun Suzuki, and Kentaro Inui. 2021. SHAPE: Shifted absolute position embedding for transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3309–3321.

Taku Kudo. 2006. MeCab: yet another part-of-speech and morphological analyzer. http://mecab.sourceforge.net.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 66–71.

Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. 2020. Understanding the difficulty of training transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5747–5763.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 220–224.

Sharan Narang, Hyung Won Chung, Yi Tay, Liam Fedus, Thibault Fevry, Michael Matena, Karishma Malkan, Noah Fiedel, Noam Shazeer, Zhenzhong Lan, Yanqi Zhou, Wei Li, Nan Ding, Jake Marcus, Adam Roberts, and Colin Raffel. 2021. Do transformer modifications transfer across implementations and applications? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5758–5773.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 160–167.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 48–53.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research (JMLR)*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2699–2712.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 86–96.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 464–468.

Sho Takase and Shun Kiyono. 2021. Rethinking perturbations in encoder-decoders for fast training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 5767–5780.

Sho Takase, Shun Kiyono, Sosuke Kobayashi, and Jun Suzuki. 2022. On layer normalizations and residual connections in transformers. *arXiv preprint arXiv:2206.00330*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems 31 (NIPS)*, pages 5998–6008.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. 2020. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 10524–10533.

Adam Mickiewicz University at WMT 2022: NER-Assisted and Quality-Aware Neural Machine Translation

Artur Nowakowski [*] [1,2] and **Gabriela Pałka** [*] [1,3] and **Kamil Guttmann** [†] [1,2] and **Mikołaj Pokrywka** [†] [1,2]
[1] Adam Mickiewicz University, Poznań, Poland
[2] Poleng, Poznań, Poland
[3] Applica.ai, Warsaw, Poland
{artur.nowakowski,gabriela.palka}@amu.edu.pl, {kamgut,mikpok1}@st.amu.edu.pl

Abstract

This paper presents Adam Mickiewicz University's (AMU) submissions to the constrained track of the WMT 2022 General MT Task. We participated in the Ukrainian ↔ Czech translation directions. The systems are a weighted ensemble of four models based on the Transformer (big) architecture. The models use source factors to utilize the information about named entities present in the input. Each of the models in the ensemble was trained using only the data provided by the shared task organizers. A noisy back-translation technique was used to augment the training corpora. One of the models in the ensemble is a document-level model, trained on parallel and synthetic longer sequences. During the sentence-level decoding process, the ensemble generated the n-best list. The n-best list was merged with the n-best list generated by a single document-level model which translated multiple sentences at a time. Finally, existing quality estimation models and minimum Bayes risk decoding were used to rerank the n-best list so that the best hypothesis was chosen according to the COMET evaluation metric. According to the automatic evaluation results, our systems rank first in both translation directions.

1 Introduction

We describe Adam Mickiewicz University's submissions to the constrained track of the WMT 2022 General MT Task. We participated in the Ukrainian ↔ Czech translation directions – a low-resource translation scenario between closely related languages.

The data provided by the shared task organizers was thoroughly cleaned and filtered, as described in section 2.

The approach described in section 3 is based on combining various MT enhancement methods, including transfer learning from a high-resource language pair (Aji et al., 2020; Zoph et al., 2016), noisy back-translation (Edunov et al., 2018), NER-assisted translation (Modrzejewski et al., 2020), document-level translation, model ensembling, quality-aware decoding (Fernandes et al., 2022), and on-the-fly domain adaptation (Farajian et al., 2017).

The results leading to the final submissions are presented in section 4. Additionally, we performed a statistical significance test with paired bootstrap resampling (Koehn, 2004), comparing the baseline solution with the final submission on the test set reference translations released by the shared task organizers. According to the automatic evaluation results based on COMET (Rei et al., 2020) scores, our systems rank first in both translation directions.

2 Data

In the initial stage of system preparation, the sentence-level data was cleaned and filtered using the OpusFilter (Aulamo et al., 2020) toolkit. With the use of the toolkit, language detection filtering based on fastText (Joulin et al., 2016) was performed, duplicates were removed, and heuristics based on sentence length were applied. In particular, we removed sentence pairs with a length ratio over 3 and long sentences (> 200 words). Then, using Moses (Koehn et al., 2007) pre-processing scripts, punctuation was normalized and non-printing characters removed. Finally, the text was tokenized into subword units using SentencePiece (Kudo and Richardson, 2018) with the unigram language model algorithm (Kudo, 2018). For Ukrainian→Czech and Czech→Ukrainian models trained from scratch, we used separate vocabularies for the source and the target language. Each vocabulary consisted of 32,000 units.

We used concatenated data from the Flores-101 (Goyal et al., 2022) benchmark (flores101-dev, flores101-devtest) for our development set, as pro-

[*] AN and GP contributed equally.
[†] KG and MP contributed equally.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 326 - 334
December 7-8, 2022 ©2022 Association for Computational Linguistics

Data type		Sentences	Corpora
Monolingual cs	available	448,528,116	News crawl, Europarl v10, News Commentary, Common Crawl, Extended Common Crawl, Leipzig Corpora
	used	59,999,553	
Monolingual uk	available	70,526,415	News crawl, UberText Corpus, Leipzig Corpora, Legal Ukrainian
	used	59,152,329	
Parallel cs-uk	available	12,630,806	OPUS, WikiMatrix, ELRC – EU acts in Ukrainian
	used	8,623,440	

Table 1: Statistics of the total available corpora and the corpora used for system training after filtering.

vided by the task organizers.

Table 1 shows statistics for the total available corpora in the constrained track and the corpora used for system training after filtering.

3 Approach

We used the Marian (Junczys-Dowmunt et al., 2018) toolkit for all of our experiments. Our model architecture follows the Transformer (big) (Vaswani et al., 2017) settings. For all model training, we used 4x NVIDIA A100 80GB GPUs.

3.1 Transfer Learning

For our initial experiments, we used transfer learning (Aji et al., 2020; Zoph et al., 2016) from the high-resource Czech→English language pair. We used only the parallel data provided by the organizers to train the model in this direction. In this case, we created a single joint vocabulary for three languages (Czech, English, Ukrainian), consisting of 32,000 units. The Czech→English model was fine-tuned for the Ukrainian→Czech and Czech→Ukrainian language directions. Our later experiments showed that there were no gains in translation quality compared with models trained from scratch using separate vocabularies for source and target languages – the upside was that the models took less time to converge during training.

3.2 Noisy Back-Translation

We used models created by the transfer learning approach to produce synthetic training data through noisy back-translation (Edunov et al., 2018). Specifically, we applied Gumbel noise to the output layer and sampled from the full model distribution. We used monolingual data available in the constrained track, which included all ~59M Ukrainian sentences after filtering and ~60M randomly selected Czech sentences.

After training the model with concatenated parallel and back-translated corpora, we replaced the training data with filtered parallel data and further fine-tuned the model. We kept the same settings as in the first training pass, training the model until it converged on the development set.

3.3 NER-Assisted Translation

Translation in domains such as news, social or conversational texts, and e-commerce is a specialized task, involving such challenges as localization, product names, and mentions of people or events in the content of documents. In such a case, it proved helpful to use off-the-shelf solutions for recognizing named entities. For Czech, the Slavic BERT model (Arkhipov et al., 2019) was used, with which entities such as persons (PER), locations (LOC), organizations (ORG), products (PRO), and events (EVT) were tagged. Due to the lack of support for the Ukrainian language in the Slavic BERT model, the Stanza Named Entity Recognition module (Qi et al., 2020) was used to detect entities in the Ukrainian text, recognizing persons (PER), locations (LOC), organizations (ORG), and miscellaneous items (MISC). With these ready-made solutions, the parallel and back-translated corpora were tagged. The named entity categories were then numbered to assign appropriate source factors to words in the text, supporting the translation process. The source factors were later transferred to subwords in a trivial way.

Source factors (Sennrich and Haddow, 2016) have previously been used to take into account various characteristics of words during the translation process. For example, morphological information, part-of-speech tags, and syntactic dependencies have been added as input to neural machine translation systems to improve the translation quality.

In the same way, it is possible to add information about named entities found in the text (Modrzejewski et al., 2020), making it easier for the model to translate them correctly. However, the AMU machine translation system does not dis-

```
Hlavní|p0 inspektor|p0 organizace|p0 RSPCA|p3 pro|p0 Nový|p2 Jižní|p2 Wales|p2
David|p1 O'Shannessy|p1 televizi|p0 ABC|p5 sdělil|p0 ,|p0 že|p0 dohled|p0 nad|p0
jatky|p0 a|p0 jejich|p0 kontroly|p0 by|p0 měly|p0 být|p0 v|p0 Austrálii|p2
samozřejmostí|p0 .|p0

_Hlavní|p0 _inspektor|p0 _organizace|p0 _R|p3 SP|p3 CA|p3 _pro|p0 _Nový|p2 _Jižní|p2
_Wales|p2 _David|p1 _O|p1 '|p1 S|p1 han|p1 ness|p1 y|p1 _televizi|p0 _A|p5 BC|p5
_sdělil|p0 ,|p0 _že|p0 _dohled|p0 _nad|p0 _ja|p0 tky|p0 _a|p0 _jejich|p0 _kontroly|p0
_by|p0 _měly|p0 _být|p0 _v|p0 _Austrálii|p2 _samozřejmost|p0 í|p0 .|p0
```

Figure 1: An example of a sentence tagged with NER source factors before and after subword encoding.

	cs				**uk**			
Category	**train-bt**	**train-parallel**	**dev**	**test**	**train-bt**	**train-parallel**	**dev**	**test**
PER	33,633,602	1,545,658	747	306	30,778,893	1,623,370	827	478
LOC	24,552,404	1,954,319	1,191	454	18,178,736	1,912,604	1,197	771
ORG	29,380,436	1,997,685	566	314	24,117,485	2,221,371	544	606
MISC	-	-	-	-	4,140,394	893,867	168	76
PRO	5,452,326	1,104,860	172	59	-	-	-	-
EVT	1,150,301	111,563	83	10	-	-	-	-

Table 2: The number of recognized named entity categories in the training, development and test data. The training data statistics are split into *train-bt*, which was created by noisy back-translation, and *train-parallel*, which is the filtered parallel training data.

tinguish between inside-outside-beginning (IOB) tags (Ramshaw and Marcus, 1995), treating the named entity tag names as a whole. Specifically, we introduce the following source factors:

- p0: source factor denoting a normal token,
- p1: source factor denoting the PER category,
- p2: source factor denoting the LOC category,
- p3: source factor denoting the ORG category,
- p4: source factor denoting the MISC category,
- p5: source factor denoting the PRO category,
- p6: source factor denoting the EVT category.

An example of a tagged sentence is shown in Figure 1.

Models were trained in two settings: concatenation and sum. In the first setting, the factor embedding had a size of 16 and was concatenated with the token embedding. In the second setting, the factor embedding was equal to the size of the token embedding (1024) and was summed with it.

As shown in Table 4, we observe an increase in the string-based evaluation metrics (chrF and BLEU) while COMET scores remain about the same. This is in accordance with Amrhein and Sennrich (2022), who show that COMET models are not sufficiently sensitive to discrepancies in named entities.

Table 2 presents the numbers of recognized named entity categories in the training, development and test data.

3.4 Document-Level Translation

Our work on document-level translation is based on a simple data concatenation method, similar to Junczys-Dowmunt (2019) and Scherrer et al. (2019).

As our training data, we use parallel document-level datasets (GNOME, KDE4, TED2020, QED), as well as synthetically created data, concatenating random sentences to match the desired input length. Specifically, we merge datasets created in the following ways as a single, large dataset:

- Curr → Curr: sentence-level parallel data,
- Prev + Curr → Prev + Curr: previous sentence given as a context,
- 50T → 50T: a fixed window of 50 tokens after subword encoding,

Netvrdím, že bakteriální celulóza jednou nahradí bavlnu, kůži, nebo jiné látky. <SEP> Ale myslím, že by to mohl být chytrý a udržitelný přírůstek k našim stále vzácnějším přírodním zdrojům. <SEP> Možná že se nakonec tyto bakterie neuplatní v módě, ale jinde. <SEP> Zkuste si třeba představit, že si vypěstujeme lampu, židli, auto, nebo třeba dům. <SEP> Má otázka tedy zní: Co byste si v budoucnu nejraději vypěstovali vy?

Figure 2: An example document consisting of five sentences separated with <SEP> tags.

- 100T → 100T: a fixed window of 100 tokens after subword encoding,
- 250T → 250T: a fixed window of 250 tokens after subword encoding,
- 500T → 500T: a fixed window of 500 tokens after subword encoding.

By concatenating such datasets, we allow the model to gradually learn how to translate longer input sequences. It is also capable of sentence-level translation. To separate sentences from each other, we introduced a <SEP> tag. An example of a document-level input sequence is shown in Figure 2. All data used to train the document-level model were tagged with NER source factors, including the back-translated data.

3.5 Weighted Ensemble

We created a weighted ensemble of four best-performing models. It consisted of the following model types:

- (A) sentence-level models trained with NER source factors (concat 16),
- (B) sentence-level model trained with NER source factors (sum),
- (C) document-level model trained with NER source factors (concat 16).

In this case, the document-level model was used only for the sentence-level translation. The optimal weights for each model were selected using a grid search method. For the specific language pairs, we used the following model and weight combinations:

- Czech → Ukrainian: $1.0 \cdot (2{\times}\text{A}) + 0.8 \cdot (\text{B}) + 0.6 \cdot (\text{C})$,
- Ukrainian → Czech: $1.0 \cdot (2{\times}\text{A}) + 0.8 \cdot (\text{B}) + 0.4 \cdot (\text{C})$.

3.6 Quality-Aware Decoding

Having the final model ensemble, we created an n-best list containing 200 translations for each sentence with beam search. Then we merged it with a second n-best list containing 50 translations for each sentence, created by a single document-level model with document-level decoding. The idea behind it was that the hypotheses produced by the document-level decoding take into account the context of surrounding sentences, which is not the case with the ensemble. This enabled the use of quality-aware decoding (Fernandes et al., 2022).

We applied a two-stage quality-aware decoding mechanism: pruning hypotheses using a tuned reranker (T-RR) and minimum Bayes risk (MBR) decoding (Kumar and Byrne, 2002, 2004), as shown in Figure 3.

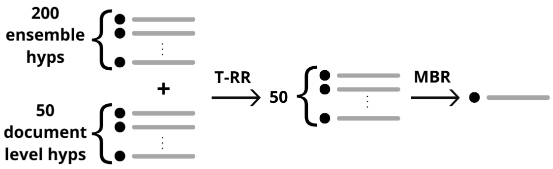


Figure 3: A two-stage (T-RR → MBR) quality-aware decoding process. 200 hypotheses generated by the ensemble are merged with 50 hypotheses generated by the document-level model. A tuned reranker is used to prune the total number of hypotheses to 50, and these are then used as input for minimum Bayes risk decoding.

First, we tuned a reranker on the development set, using as features NMT model scores, as well as existing QE models based on TransQuest (Ranasinghe et al., 2020) and COMET (Rei et al., 2020), which are based on Direct Assessment (DA) (Graham et al., 2013) scores or MQM (Lommel et al., 2014) scores. Specifically, we used:

- model ensemble log-likelihood $\log p_\theta(y|x)$ scores,

- TransQuest QE model trained on DA scores (`monotransquest-da-multilingual`),
- COMET QE model trained on MQM scores (`wmt21-comet-qe-mqm`),
- COMET QE model trained on DA scores (`wmt21-comet-qe-da`).

We tuned the feature weights to maximize the COMET reference-based evaluation metric value using MERT (Och, 2003).

After tuning the reranker, we used it to prune the n-best list from 250 to 50 hypotheses per input sentence. The resulting n-best list was used for minimum Bayes risk decoding, using the COMET reference-based metric as the utility function. Minimum Bayes risk decoding seeks, from the set of hypotheses, the hypothesis with the highest expected utility.

$$\hat{y}_{\text{MBR}} = \arg\max_{y \in \bar{\mathcal{Y}}} \underbrace{\mathbb{E}_{Y \sim p_\theta(y|x)}[u(Y, y)]}_{\approx \frac{1}{M}\sum_{j=1}^{M} u(y^{(j)}, y)} \qquad (1)$$

Equation 1 shows that the expectation can be approximated as a Monte Carlo sum using model samples $y^{(1)}, \ldots, y^{(M)} \sim p_\theta(y|x)$. In practice, the translation with the highest expected utility can be chosen by comparing each hypothesis $y \in \bar{\mathcal{Y}}$ with all other hypotheses in the set.

The described two-stage quality-aware decoding process allowed us to further optimize our system for the COMET evaluation metric, which has been shown to have a high correlation with human judgements (Kocmi et al., 2021).

3.7 Post-Processing

The final step involved post-processing. We applied the following post-processing steps for each best obtained translation:

- transfer of emojis from the source to the translation using word alignment based on SimAlign (Jalili Sabet et al., 2020),
- restoration of quotation marks appropriate for a given language,
- restoration of capitalization (e.g. if the source sentence was fully uppercased),
- restoration of punctuation, exclamation and question marks (if a source sentence ends with such a mark, we make the translation do likewise),
- replacement of three consecutive dots with an ellipsis,
- restoration of bullet points and enumeration (e.g. if the source sentence starts with a number or a bullet point),
- deletion of consecutively repeated words.

Approach	**Sim. score**	**COMET**	**chrF**
Baseline	-	0.8322	0.5263
Default	0.4	0.8316	0.5260
Best-334	0.19	0.8322	0.5259
Best-133	0.25	0.8323	0.5262

Table 3: Results of the on-the-fly adaptation method on the development set. The *default* approach is based on Farajian et al. (2017). However, only 11 sentence pairs were found in this scenario. The experiments denoted as *best-334* and *best-133* used the learning rate values of 0.002 and 10 epochs. In our development set containing 2009 sentence pairs, 334 matching sentences were found in *best-334* and 133 in *best-133*.

3.8 On-The-Fly Domain Adaptation

The General MT Task tests the MT system's performance on multiple domains. Therefore, we investigated the possibility of improving our translation system with the on-the-fly domain adaptation method.

This experiment was based on Farajian et al. (2017). Our idea was to retrieve similar sentences from the training data for each input sentence and to fine-tune the model on their translations. After the translation of a single sentence is complete, the model is reset to the original parameters. We used Apache Lucene (McCandless et al., 2010) as our translation memory to search for similar sentences. We indexed all of the training data and used the Marian dynamic adaptation feature. We compared the translation quality with and without the retrieved context. The experiments were carried out with a different similarity score used to choose similar sentence pairs for the fine-tuning process. We empirically modified the learning rate and the number of epochs to find optimal values that improved the translation quality.

Table 3 shows the results of the aforementioned experiments on the full development set. We found

System		uk→cs			cs→uk		
		COMET	chrF	BLEU	COMET	chrF	BLEU
Baseline (transformer-big)		0.8622	0.5229	24.29	0.7818	0.5175	22.64
+back-translation		0.9053	0.5309	25.41	0.8356	0.5280	23.14
+ner	concat 16	0.9003	0.5314	25.62	0.8362	0.5309	24.28
	sum	0.8991	0.5323	25.87	0.8421	0.5302	23.91
+fine-tune	concat 16	0.9021	0.5344	25.94	0.8387	0.5330	24.51
	sum	0.8990	0.5357	25.99	0.8456	0.5321	24.24
+ensemble		0.9066	0.5376	**26.36**	0.8522	0.5373	**24.85**
+quality-aware		0.9874	0.5376	25.42	0.9238	0.5384	24.50
+post-processing		**0.9883**	**0.5392**	25.89	**0.9240**	**0.5388**	24.63
Document-level	sent-level dec.	0.8942	0.5326	25.47	0.8350	0.5289	23.92
	doc-level dec.	0.8920	0.5324	25.44	0.8356	0.5297	23.78

Table 4: Results of COMET, chrF and BLEU automatic evaluation metrics on the concatenated datasets flores101-dev and flores-101-devtest. ChrF and BLEU metrics were computed with sacreBLEU. Document-level model evaluation includes added back-translation, NER source factors (concat 16) and fine-tuning.

System	uk→cs			cs→uk		
	COMET	chrF	BLEU	COMET	chrF	BLEU
Baseline (transformer-big)	0.8315	0.5627	31.79	0.8008	0.5849	31.43
Final submission	**1.0488**	**0.6066**	**37.03**	**0.9944**	**0.6153**	**34.74**

Table 5: Results of COMET, chrF and BLEU automatic evaluation metrics on the test set. ChrF and BLEU metrics were computed with sacreBLEU. The final submission results are statistically significant ($p < 0.05$).

that only a small number of sentences in the training data were similar to those present in the development set. The results showed that tuning the model on similar sentences from the training data did not significantly improve translation quality. In the end, we decided not to use this method in our WMT 2022 submission.

4 Results

The results of our experiments are presented in Table 4. We evaluated our models with the COMET[1] (Rei et al., 2020), chrF (Popović, 2015) and BLEU (Papineni et al., 2002) automatic evaluation metrics. ChrF and BLEU scores were computed with the sacreBLEU[2,3] (Post, 2018) tool. We also include scores for the document-level model. In this case, the scores include improvements added by back-translation, NER source factors and fine-tuning. The document-level evaluation was split into sentence-level decoding and document-level decoding. In the first scenario, the model translates a single sentence at a time, which is not different from a sentence-level model. In the second scenario, the model translates concatenated chunks of at most 250 subword tokens at a time.

We found that the largest gain in the COMET value was achieved due to the quality-aware decoding method, at the cost of BLEU value. The chrF value remained the same in the Ukrainian→Czech translation direction, while it increased slightly in the Czech→Ukrainian direction. As discussed in section 3.3, the inclusion of NER source factors helped the model with the translation of named entities, which is not well reflected in the COMET value, as this metric is not sufficiently sensitive to discrepancies in named entities (Amrhein and Sennrich, 2022).

Table 5 shows results for our final submissions compared with the baseline. We performed a statistical significance test with paired bootstrap resampling (Koehn, 2004), running 1000 resampling trials to confirm that our submissions are statistically significant ($p < 0.05$).

5 Conclusions

We describe Adam Mickiewicz University's (AMU) submissions to the WMT 2022 General

[1] COMET scores were computed with the `wmt20-comet-da` model.

[2] BLEU signature: nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.0

[3] chrF signature: nrefs:1|case:mixed|eff:yes|nc:6|nw:0|space:no|version:2.0.0

MT Task in the Ukrainian ↔ Czech translation directions. Our experiments cover a range of MT enhancement methods, including transfer learning, back-translation, NER-assisted translation, document-level translation, weighted ensembling, quality-aware decoding, and on-the-fly domain adaptation. We found that using a combination of these methods on the test set leads to a +0.22 (26.13%) increase in COMET scores in the Ukrainian→Czech translation direction and a +0.19 (24.18%) increase in the Czech→Ukrainian direction, compared with the baseline model. According to the COMET automatic evaluation results, our systems rank first in both translation directions.

References

Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. In neural machine translation, what does transfer learning transfer? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7701–7710, Online. Association for Computational Linguistics.

Chantal Amrhein and Rico Sennrich. 2022. Identifying weaknesses in machine translation metrics through minimum Bayes risk decoding: A case study for COMET. *arXiv preprint arXiv:2202.05148*.

Mikhail Arkhipov, Maria Trofimova, Yuri Kuratov, and Alexey Sorokin. 2019. Tuning multilingual transformers for language-specific named entity recognition. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 89–93, Florence, Italy. Association for Computational Linguistics.

Mikko Aulamo, Sami Virpioja, and Jörg Tiedemann. 2020. OpusFilter: A configurable parallel corpus filtering toolbox. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 150–156. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

M. Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *Proceedings of the Second Conference on Machine Translation*, pages 127–137, Copenhagen, Denmark. Association for Computational Linguistics.

Patrick Fernandes, António Farinhas, Ricardo Rei, José De Souza, Perez Ogayo, Graham Neubig, and Andre Martins. 2022. Quality-aware decoding for neural machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1396–1412, Seattle, United States. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233, Florence, Italy. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the*

2004 Conference on Empirical Methods in Natural Language Processing, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Shankar Kumar and William Byrne. 2002. Minimum Bayes-risk word alignments of bilingual texts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 140–147. Association for Computational Linguistics.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 169–176, Boston, Massachusetts, USA. Association for Computational Linguistics.

Arle Lommel, Hans Uszkoreit, and Aljoscha Burchardt. 2014. Multidimensional quality metrics (mqm): A framework for declaring and describing translation quality metrics. *Revista Tradumàtica: tecnologies de la traducció*, (12):455–463.

M. McCandless, E. Hatcher, and O. Gospodnetić. 2010. *Lucene in Action*. Manning Pubs Co Series. Manning.

Maciej Modrzejewski, Miriam Exel, Bianka Buschbeck, Thanh-Le Ha, and Alexander Waibel. 2020. Incorporating external annotation to improve named entity translation in NMT. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 45–51, Lisboa, Portugal. European Association for Machine Translation.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics – Volume 1*, ACL '03, page 160–167, USA. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. TransQuest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5070–5081, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Yves Scherrer, Jörg Tiedemann, and Sharid Loáiciga. 2019. Analysing concatenation approaches to document-level NMT in two different domains. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 51–61, Hong Kong, China. Association for Computational Linguistics.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine*

Translation: Volume 1, Research Papers, pages 83–91, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

Evaluating Corpus Cleanup Methods in the WMT’22 General Translation Task

Marilena Malli
Department of Informatics and Telecommunications,
University of Athens
Ilissia, 15784, Greece
mallimariaeleni@gmail.com

George Tambouratzis
ILSP, Athena R.C.
6 Artemidos Str.,
Maroussi, 15125, Greece
giorg_t@athenarc.gr

Abstract

This paper describes the LT’22 team’s constrained submission to the WMT General Machine Translation task. NMT transformer-based systems have been implemented using only the WMT’22 released parallel corpora, without using any pre-trained models. Two language pairs have been tackled, namely German to English and German to French. Emphasis was placed on removing the noisy sections of parallel corpora where the degree of parallelism is very limited, for which a publicly-available tool was-used. Comparative results are reported with baseline systems.

1 Introduction

This submission presents the contribution of the LT’22 team to the WMT22: General MT Task. It focuses on studying the effectiveness of cleaning tools when these are applied to real-world parallel corpora, to eliminate noisy sections and improve the resulting NMT systems.

Traditionally, parallel corpora are used as the primary data source for machine translation (MT) models. The development of MT has been aided by the availability of extensive parallel corpora. The majority of these data have several areas of reduced parallelism and are usually characterized as imperfect or noisy. The use of noisy data may result in a neural machine translation model being inadequately prepared. Researchers (e.g. (Koehn and Knowles, 2017) (Khayrallah and Koehn, 2018)) have reported that neural machine translation models are much more affected by noisy data than statistical machine translation models.

A number of software packages to implement noise-removal from parallel corpora have been implemented and released to the community. These include publicly available tools such as qe-clean (Denkowski)[1], as well as Zipporah (Xu and Koehn, 2017). (Zariņa et al., 2015) have used a combination of alignment-indicating features to clean corpora. For cleaning large-scale corpora in multilingual setups, a cosine-distance metric has been proposed (Schwenk and Li, 2018). Finally, the suite of the paired Bifixer and Bicleaner software tools (Ramírez-Sánchez et al., 2020) has been proposed for parallel corpora cleaning purposes, with Bifixer implementing restorative cleaning and Bicleaner providing the ability to remove sentences with very low parallelism in the parallel corpus.

For the experiments reported here, two language pairs have been chosen, namely German-to-English (denoted as De-to-En) and German-to-French (denoted as De-to-Fr). Compared with other systems reported in WMT, our NMTs have a couple of identifying features: (1) the use of a fully-constrained setup with respect to WMT’22 rules and (2) the setting of a relatively low threshold to the allowed training epochs, in an effort to comply to a setup with limited computational resources. Whilst our translation systems are not as accurate as they could be if more epochs were allowed, it was decided to adopt an approach that is more realistic when training resources are not unlimited.

To implement the LT’22 participation to the WMT’22 shared task work, we used the following three software packages: (i) the Marian NMT Toolkit (Version: v1.11.5), which was used for the training of the neural machine translation models and (ii) Bifixer and (iii) Bicleaner, which were used in order to correct and clean our data.

Regarding the structure of the paper, in the second section the selection of data on which to train the translation systems is reported. In the third section, the method used to carry out all essential experiments is detailed. In the fourth section, the corpus-cleaning tools are analyzed. In the fifth section the translation systems and their parameters are reported. The sixth section is devoted to details related to experiments. Finally, we review the

[1] https://github.com/mjdenkowski/qe-clean

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 335 - 341
December 7-8, 2022 ©2022 Association for Computational Linguistics

findings of this series of experiments and examine potential future research directions.

2 Training Data

Our experiments involve comparing the translation outputs for a series of NMT models for two language pairs: German-to-English (denoted as De-to-En) and German-to-French (denoted as De-to-Fr). It should be noted that for these two language pairs no pretrained models for either Bifixer or Bicleaner are available at the respective repository. All the NMT models reported here are trained using only the parallel training data specified by WMT'22, and no monolingual training data are used. In-training validation has been performed using the development data recommended in WMT'22, whilst for evaluating the trained NMT systems (developed prior to the release of WMT'22 test data), the relevant test data from WMT'20 were used. Moreover, the translations submitted at the WMT22 shared task have been produced using the test data released by WMT'22.

3 Methodology

The aim of our experiments has been to evaluate methods for cleaning-up a parallel corpus and to determine if their use leads to MT systems that generate more accurate translations. For each language pair, baseline NMT models have been trained from raw (i.e. unfiltered) parallel training corpora as specified by WMT'22, while the additional NMT models have been trained with corpora subjected to a special cleaning process via the Bifixer and Bicleaner suite (Ramírez-Sánchez et al., 2020). It should be mentioned that the Bicleaner repository[2] doesn't include pre-trained classifiers for the above language pairs; consequently we trained probabilistic dictionaries in order to produce new models. An added benefit of this choice is that no pre-trained model was used to develop our NMT systems, and thus the submitted systems reviewed here are constrained.

The fundamental differences between the NMT models produced are mainly related to the quality and quantity of the training data, while there are no differences in the training parameters or in the setup of the deep neural network architectures (unless otherwise noted in the experimental section). By doing so, it is possible to safely draw conclusions about the amount of computational resources required while also examining and comparing the translation outputs using automatic assessment methods. The following were the driving factors behind the experiments reported here:

- Using the Bifixer/Bicleaner tool in other language pairs for which they have not been used to date, in order to observe their effectiveness in a different real-world scenario.

- The comparison of the results of cleaned as well as raw parallel corpora, automatically as well as manually.

- The study of the effectiveness of translation models produced with limited computing resources (Arase et al., 2021).

4 Cleaning Parallel Corpora

4.1 Bifixer

The first tool that was used in the translation pipeline is Bifixer, which undertakes to correct some very specific errors that publicly available parallel corpora usually present. Bifixer implements restorative cleaning of imperfect parallel data, working towards fixing the content and preserving unique parallel sentences before filtering out the noise (Ramírez-Sánchez et al., 2020). The steps followed involve empty side removal, character fixing, orthography fixing, re-splitting, duplicates identification. In order to apply Bifixer, we used the recommended default parameter values, without changes, and noted an improvement in the quality of the parallel corpora.

100 random sentence pairs were examined in order to ascertain the effectiveness of Bifixer. After using of the aforementioned tool, fewer noisy data were observed. Better sentence segmentation, fewer typographical errors and fewer extremely short and big sentences were the most notable modifications.

4.2 Bicleaner

Continuing the corpus-cleaning process, we proceeded to the next tool, Bicleaner. This tool filters parallel corpora in order to distinguish the noisiest sentences and then remove them to create a cleaner corpus.

In order to use Bicleaner we need to have an already trained classifier. Hence, we initiated the

[2] https://github.com/bitextor/bicleaner-data/releases/tag/v1.5

Bicleaner training process, following the steps described in the official github page [3].

The assembly of a big corpus consisting of about 10 M sentences was our first concern. In order to avoid bias, the sentences were chosen to be different from those used to train Marian NMT models. The training data went through a simple preprocessing which consists of the following steps; detokenization in case of already tokenized corpora; then tokenization of all sentences. As the same tokenization method will be used during Bicleaner running, and the parallel data needs to be aligned in both directions, we used MGIZA++ (Gao and Vogel, 2008). Another software package we used was Moses[4], which is utilized for tokenization as well as the construction of probabilistic dictionaries in combination with MGIZA++. Following this process, two probabilistic dictionaries are constructed, one for each translation direction.

The next step was to create word frequency files. Two folders are needed, for the source language and the target language. To build these two folders we needed two large monolingual corpora. Besides, ideally a very clean corpus of about 100K sentences is required, though such clean data are not readily available. According to the recommendations in github in this case the data can be cleaned by using Bifixer and the Bicleaner Hardrules, which given a parallel corpus, seek to identify evident noisy sentence pairs (Sánchez-Cartagena et al., 2018).

After gathering the aforementioned material, the final step is the training of the Bicleaner. Furthermore, to create the character language models, we utilize the KenLM software package (Heafield, 2011). Via these steps, a trained classifier ready for use in pre-processing was obtained.

5 Training the NMT Systems

For training neural machine translation models, we chose the Marian NMT toolkit (Junczys-Dowmunt et al., 2018). Marian was developed to allow rapid training and translation speed, to facilitate the standardization of research work. All the models we trained adopted the architecture of a sequence-to-sequence transformer with 8 attention heads and 6 layers in both the encoder and decoder, thus largely adhering to the standard transformer configuration from (Vaswani et al., 2017). We also decided to set a specific limit to the number of training epochs to avoid lengthy training sessions, aiming to economize as far as possible on valuable computing resources, as per the recent ACL recommendation for efficient computing (Arase et al., 2021).

The transformer is characterized as innovative and uncomplicated (Vaswani et al., 2017). In our experiments, we activated the dropout mechanism, which is a widely adopted regularisation technique in NMT.

When training our NMT systems, we opted to use the SentencePiece tokenizer, which has the ability to train subword models straight from unprocessed data (Kudo and Richardson, 2018). The vocabulary size was set to 32000 and the range for the batch size was from 64 to 100. For the workspace size we used a variable value across our experiments, as the size of the training corpora varied due to the Bicleaner filtering. As suggested by the Marian developers, the workspace was adapted via a number of trial runs at the start of the training process, to maximise the throughput of training sentences per time unit. The other main parameter choices for the transformer models are shown in Table 1. Moreover, the full command used for training is presented in Table 3.

Translation Systems	
encoder/decoder depth	6
beam size	6
layer normalization	yes
exponential smoothing	yes
mormalize factor	0.6
early stopping	5
transformer dropout	1
transformer dropout attention	1
dropout-rnn	0.2
dropout-src	0.1
dropout-trg	0.1

Table 1: Main parameters of the transformer architecture used.

6 Experiments

6.1 Experimental setup

As discussed above, the training data used to implement all the reported experiments were limited to the parallel corpora released for WMT22 for the two language pairs German-French and German-English. For the baseline systems the text

[3] https://github.com/bitextor/bicleaner/wiki/How-to-train-your-Bicleaner

[4] http://www2.statmt.org/moses/

corpora of the respective language pair were used as released, without any pre-processing or noise-removal. Contrariwise, the remaining experiments were carried out using the aforementioned cleaning tools. After applying Bicleaner, the content of the parallel corpus remains the same, however an extra column is added where the parallelism ratings that the classifier assigned to each pair of parallel sentences are stored. Based on this column, sentence pairs rated below a threshold are discarded. Although 0.5 is suggested as a desirable threshold in relevant literature, we chose to examine other thresholds. For this reason, we tested different threshold values within the range from 0.4 to 0.7 to to discover whether changes in this parameter affect the translation accuracy of neural machine translation models. Table 2 provides details regarding the number of sentences that are retained in the parallel corpus following each application of Bicleaner.

Corpora(de-en)	Sentences
baseline_corpus.de_en	∼2.800.000
0.7_corpus.de_en	∼1.100.000
0.6_corpus.de_en	∼1.500.000
0.5_corpus.de_en	∼1.600.000
0.4_corpus.de_en	∼1.700.000
Corpora(de-fr)	**Sentences**
baseline_corpus.de_fr	∼18.000.000
0.7_corpus.de_fr	∼7.800.000

Table 2: Volume of data before and after the cleaning process.

6.2 Computer resources

For the experiments presented here a workstation was used, equipped with a single Nvidia GeForce RTX-3090 GPU, and an Intel i9-11900 CPU with 32 GB of memory. The first two tools were run on the CPU whilst the NMT models training via Marian involved predominantly the GPU. For all experiments where execution times are reported, these times are obtained with the workstation running exclusively the reported process.

6.3 Experimental results

At this point, we will review the Marian NMT training results. In Table 4 the BLEU scores during experimental process are presented. Additionally in Table 5, the WMT22 results of the automatic evaluation metrics can be found. Regarding the German-English language pair, we can observe that the baseline system has the highest score. Implementing the cleaning steps and increasing the threshold, the size of training data gets smaller and smaller, as can be seen in Table 1. Since the size of the initial data was not very big, the decrease of the data may well affect the efficiency of the models.

Regarding the German-French language pair, the best score is observed in the model trained on cleaned data. As is mentioned in a related study (Ramírez-Sánchez et al., 2020), it has been observed that the Bifixer/Bicleaner tools work better on big data. In this case the number of the sentences continues to be adequate even after the cleaning process.

7 Conclusions and Future Work

In this paper, we have presented our submission to the WMT22: General MT Task. In order to rectify and filter noisy sentences from the corpora recommended by WMT'22, we have applied two cleaning approaches for the parallel corpus. After experimenting with various categorization criteria, we created seven distinct parallel corpora. We discovered that as expected, thoroughly cleaned corpora require fewer computer resources, as a large number of sentences are removed. Additionally, we noticed that differences in the BLEU score across cleaned corpora are relatively small.

Our main submissions to the shared task were two, one for each language pair. Regarding the language pair German to English, the highest quality translation result was obtained by training a transformer model using the raw baseline corpus, and thus the use of Bifixer/Bicleaner did not lead to an improvement. The best result was obtained for the language pair German to French by training a transformer model using the bifixed and bicleaned parallel corpus with a threshold of 0.5.

In upcoming research, the Back Translation technique is planned to be utilized in order to expand the size of the training data, since the size of the sentence pairs is reduced after the cleaning procedure. The translations that emerged from the aforementioned experimental process could be filtered and reused so as to train the NMT system with bigger and cleaner parallel corpora.

In the future, it would be highly interesting to develop probabilistic dictionaries with more than 10 M parallel sentences as well as to train the Bicleaner in more than 100 K parallel sentences. Additionally, we want to use these methods on even more information about the language pairs we previously stated. In order to achieve an even cleaner corpus, it would also be quite fascinating to investigate comparatively other cleaning techniques such as those reported in the introduction.

A final direction for future work would be to use larger models such as the Big Transformer (Vaswani et al., 2017) to see if for this architecture the effect of pre-filtering with Bifixer/Bicleaner will be more marked, and what the trade-off between the improvement in translation quality and the increased training time would be.

Limitations One potential limitation of the present work is the relatively limited range and number of Bicleaner thresholds tested, though the values include both the recommended and default values. Another limitation concerns the use of a single architecture, whilst ideally a second architecture (such as the Transformer-Big configuration of (Vaswani et al., 2017)) could be used. Finally, a comparison with other corpus-cleaning methods would be desirable, though such work is beyond the scope of the present work.

Ethics Statement The present work is not expected to have any effect on ethical issues and to the authors' best knowledge complies with the ACL Ethics Policy.

Acknowledgements The current research was carried out in the context of a master's thesis implemented by the first author in the course "Language Technology" organised by the National and Kapodistrian University of Athens and ILSP/Athena R.C.. We would like to acknowledge the assistance of the WMT'22 organisers and especially Tom Kocmi during the implementation of the experiments and the submission of our results.

References

Yuki Arase, Phil Blunsom, Mona Diab, Jesse Dodge, Iryna Gurevych, Percy Liang, Colin Raffel, Andreas Rücklé, Roy Schwartz, Noah A. Smith, and Emma Strubell. 2021. Efficient nlp policy document. In *Efficient NLP Policy Document, Association of Computational Linguists, November*.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57, Columbus, Ohio. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón, and Sergio Ortiz Rojas. 2020. Bifixer and bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298, Lisboa, Portugal. European Association for Machine Translation.

Víctor M. Sánchez-Cartagena, Marta Bañón, Sergio Ortiz-Rojas, and Gema Ramírez. 2018. Prompsit's submission to WMT 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 955–962, Belgium, Brussels. Association for Computational Linguistics.

Holger Schwenk and Xian Li. 2018. A corpus for multilingual document classification in eight languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz

Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2945–2950, Copenhagen, Denmark. Association for Computational Linguistics.

Ieva Zariņa, Pēteris Ņikiforovs, and Raivis Skadiņš. 2015. Word alignment based parallel corpora evaluation and cleaning using machine learning techniques. In *Proceedings of the 18th Annual Conference of the European Association for Machine Translation*, pages 185–192, Antalya, Turkey.

Translation Systems

```
~/marian/build/marian –model modelsname.npz \
- -vocabs modelsname/vocabsname.deen.spm modelsname/vocabsname.spm \
- -type transformer - -transformer-heads 8 - -train-sets ~/corpus.srl \
~/corpus.trl - -disp-freq 100 - -mini-batch-fit - -workspace 21000 \
- -layer-normalization - -exponential-smoothing \
- -sentencepiece-alphas 0.2 0 \
- -dim-vocabs 32000 32000 \
- -after-epochs 21 - -dropout-rnn 0.2 - -dropout-src 0.1 - -dropout-trg 0.1 - -valid-metrics cross-entropy \
- -valid-sets ~/dev.srl ~/dev.trl - -valid-freq 10000 \
- -beam-size 6 - -normalize=0.6 - -early-stopping 5 \
- -cost-type=ce-mean-words - -max-length 200 - -save-freq 10000 \
- -overwrite - -keep-best - -log ~/transformer.log \
- -valid-log ~/transformer_valid.log \
- -enc-depth 6 - -dec-depth 6 - -learn-rate 0.0001 \
- -lr-warmup 8000 - -lr-decay-inv-sqrt 8000 - -lr-report \
- -seed 1 - -label-smoothing 0.1
```

Table 3: An example command used in order to train NMT systems with Marian.

Data	Cleaning Method	Threshold	BLEU	Training Time
System1.de-en	None(raw data)	-	17.4	~66h
System2.de-en	Bifixer/Bicleaner	0.4	22.7	~26h
System3.de-en	Bifixer/Bicleaner	0.5	23.2	~26h
System4.de-en	Bifixer/Bicleaner	0.6	24.1	~19h
System5.de-en	Bifixer/Bicleaner	0.7	23.3	~15h
System1.de-fr	None(raw data)	-	26.3	~92h
System2.de-fr	Bifixer/Bicleaner	0.7	27.6	~74h

Table 4: BLEU scores on WMT20 test during the development process.

Data	Cleaning Method	Threshold	BLEU	chrF	COMET-A	COMET-B
System1.de-en*	None(raw data)	-	26.0	0.5	25.6	33.3
System2.de-en	Bifixer/Bicleaner	0.4	24.3	0.5	N/A	N/A
System3.de-en	Bifixer/Bicleaner	0.5	25.3	0.5	N/A	N/A
System4.de-en	Bifixer/Bicleaner	0.6	24.9	0.5	N/A	N/A
System5.de-en	Bifixer/Bicleaner	0.7	24.0	0.5	N/A	N/A
System1.de-fr*	None(raw data)	-	24.4	0.5	N/A	N/A
System2.de-fr	Bifixer/Bicleaner	0.7	28.3	0.5	10.4	54.4

Table 5: Cleaning method, WMT22 automatic scores and training time for all submitted NMT systems. *Systems defined as primaries.

PROMT Systems for WMT22 Shared General Translation Task

Alexander Molchanov, Vladislav Kovalenko & Natalia Makhamalkina
PROMT LLC
17E Uralskaya str. building 3, 199155,
St. Petersburg, Russia
First.Last@promt.ru

Abstract

This paper describes the PROMT submissions for the WMT22 Shared General Translation Task. This year we participated in four directions of the Shared Translation Task: English to Russian, English to German and back, and Ukrainian to English. All our models are trained with the MarianNMT toolkit using the transformer-big configuration. We use BPE for text encoding, all of our models are unconstrained. We achieve competitive results according to automatic metrics in all directions.

1 Introduction

The WMT Shared General Translation Task is an annual event where different companies and researchers build and test their systems on the test sets provided by the organizers. This year the Task has shifted from news to the general domain. We participate in four directions: English to Russian, English to German and back, and Ukrainian to English. We build the transformer-big models for the first time. We also explore new data filtering techniques, data preparation and model training strategies.

The rest of the paper is organized as follows: in Section 2 we describe in detail the systems we submitted to the Shared Task. In Section 3 we present and discuss the results. We conclude the paper in Section 4 with discussion for possible future work.

2 Systems overview

All of our WMT22 submissions are MarianNMT-trained (Junczys-Dowmunt et al., 2018) transformer-big (Vaswani et al., 2017) systems. We use the OpenNMT toolkit (Klein et al., 2017) version of byte pair encoding (BPE) (Sennrich et al., 2016b) for subword segmentation. Our BPE models are case-insensitive, we use special tokens in the source and target sides to process case (see Molchanov (2019) for details).

All of the systems are unconstrained, i.e. we use all data provided by the WMT organizers, all publicly available data and some private data crawled from different web-sources.

This year we use the dual conditional cross-entropy (Junczys-Dowmunt, 2018) method for data filtering. We extend the method as proposed by the author and build neural language models for both source and target languages.

We also augment our training data with two types of synthetic data: 1) back-translations (Sennrich et al., 2016a) and 2) synthetic data with placeholders as described in Pinnis et al. (2017). The back-translations are obtained using the previous versions of our NMT models which are baseline transformers trained with less data (and without some up-to-date data like the news 2021 corpora from statmt.org). We also tag all our synthetic data with special tokens at the beginning of the source sentences as described in Caswell et al. (2019).

All models are trained with guided alignment which is used at translation time to handle named entities and document formatting. We obtain

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 342 - 345
December 7-8, 2022 ©2022 Association for Computational Linguistics

	German-English		Russian-English		Ukrainian-English	
	#sent	#tokens EN	#sent	#tokens EN	#sent	#tokens EN
WMT+OPUS	148.0	4000.1	37.4	690.9	24.8	566.7
Private	8.1	106.8	30.2	542.2	0.5	5.8
Total	156.1	4106.9	67.6	1233.1	25.3	572.5

Table 1: Statistics for the filtered human parallel data in millions of sentences (#sent) and tokens (#tokens) for three language pairs. WMT stands for the data available for the News Task on the statmt.org/wmt22 website; OPUS is the data from the OPUS website apart from the data available for the News Task; Private stands for private company data.

alignments using the `fast-align` (Dyer et al., 2013) tool.

The data statistics for different language pairs are presented in Table 1.

The details regarding different directions can be found in the next Section.

2.1 Data preparation

There are several stages in our data preparation pipeline. These are mostly common filtering techniques. The main stages of the pipeline are:

- Basic filtering
  This includes some simple length-based and source-target length ratio-based heuristics, removing tags, lines with low amount of alphabetic symbols etc. We also remove lines which appear to be emails or web-addresses and duplicates.

- Language identification
  The algorithm is a fairly simple ensemble of three tools: `pycld2`[1], `langid` (Lui and Baldwin, 2012), `langdetect`[2]. For large monolingual corpora we use only pycld2.

- Bicleaner filtering
  We use the bicleaner (Ramírez-Sánchez et al., 2020) tool to filter parallel data. We discard all sentence pairs with the score threshold <= 0.3.

- Scoring with NMT models
  We finally score all parallel data and back-translations with our intermediate models to discard non-parallel sentence pairs and bad synthetic translations.

- Dual conditional cross-entropy filtering
  This year we use this algorithm for the first time. We apply it to the English-German language pair.

2.2 English-Russian

The English—Russian system was trained in two steps. First, we build the baseline model on all available data. Second, we fine-tune the model on data of high quality. Specifically, we totally remove the ParaCrawl, UN and OpenSubtitles corpora and fine-tune the model using the remains of the human data mixed with the back-translations of the news corpora (2020, 2021) from statmt.org. This approach shows good results according to automatic metrics and general translation quality. The reason for doing this is that we aim for our models to be used mostly for translation of news and formal texts like various types of documents.
The system was trained with separate vocabularies, the sizes of the BPE models are 24k for the source side and 48k for the target side.

2.3 English-German and German-English

Both models were trained with the same joint vocabulary, the BPE model size is 32k. We use all available human data. We apply basic filtering for some data which we believe to be clean (e.g. private data and high-quality open-source corpora like News-Commentary). The rest of the data is filtered with the modified dual conditional cross-entropy filtering algorithm. We noticed that using only the news corpora as general for filtering as described in Junczys-Dowmunt (2018) results in the fact that the data shifts towards the news domain. For example, a perfectly fine sentence

[1] https://pypi.org/project/pycld2/

[2] https://pypi.org/project/langdetect/

pair related to the IT domain may receive low scores from News models. Therefore, we try to build a general good quality corpus comprising different domains (news, IT, technical data etc.). We do not include colloquial corpora into these general corpora because we intend for our models to be used for translating mostly formal text, be it news, formal letters or technical documents. We set the threshold for the filtering score at 0.1. Thus, we discard around 60-70% of the original data.

2.4 Ukrainian-English

We use a lot of synthetic data for this model. We decided that we could pivot the Ukrainian-English model through our Ukrainian-Russian and English-Russian data and systems. We translate the Russian side of the English-Russian data to Ukrainian and use it as synthetic data for the final model.

The Ukrainian-Russian model is a transformer-base unconstrained model. It was built jointly to translate from Ukrainian into Russian and back. We use all available parallel data and back-translations of the news and Wikipedia corpora. Although this is a transformer-base model, the Ukrainian-Russian language pair is relatively easy for the model to learn properly and achieve very good results in. Thus, we made an assumption that even the big model would benefit from this synthetic data given the fact that the Ukrainian-English is not a high-resource language pair.

To see how much we benefit specifically from using the transformer-big architecture in addition to the synthetic data from the Russian-English pair we also build a transformer-base model for this language pair.

3 Results and discussion

The results are presented in Table 2.

As we can see, we clearly outperform our baselines (i.e. previous versions of the models). The gains we observe, however, are not that large.

We notice that our submitted models have some problems with translation of colloquial content compared to the previous versions. This can be explained by our data preparation scheme. As we have already mentioned above, we want our models to translate formal text better and thus ‘sacrifice’ colloquial data. The examples of such degradations are presented in Table 3. The first example illustrates the problem when short colloquial segments are left untranslated. We think there are two major reasons for that: 1) the fine-tuned model has partially ‘forgotten’ how to translate colloquial speech; 2) there are many technical and IT-related texts in the fine-tuning data where large constructions (e.g. model or software program names) are left untranslated. Two other examples illustrate bad choice of meaning for specific words from the fine-tuned translation model (‘screwed’ is translated literally as if the kid was attached to something with a screwdriver; ‘кредит’ is a word from the financial domain which is inappropriate in this context).

System	**BLEU**	**chrF**	**COMET**
English-Russian			
Model2021	29.1	52.5	0.54
Model2022	**30.6**	**53.8**	**0.60**
English-German			
Model2021	45.3	62.8	0.49*
Model2022	**49.0**	**65.3**	**0.55***
German-English			
Model2021	47.3	62.3	0.51*
Model2022	**49.1**	**63.8**	**0.55***
Ukrainian-English			
Model2021	38.6	60.4	0.44
Model2022 base	39.7	61.3	0.46
Model2022	**41.2**	**62.6**	**0.49**

Table 2: Results for different systems and directions. The submitted systems are marked in bold. The starred scores are averaged scores over two references provided by the organizers. Model2021 stands for our previous versions of the systems which we consider the baseline. Model2022 base stands for the transformer-base configuration of the 2022 model.

Source text	**Model2021**	**Model2022**
You meet me	Встретишь меня	You meet me
And this kid is screwed.	И этот парень облажался.	И этот пацан прикручен.
I don't have enough credits to graduate.	У меня недостаточно баллов, чтобы закончить школу.	У меня недостаточно кредитов, чтобы получить высшее образование.

Table 3: Examples of translation degradation for colloquial content in the English-Russian direction. Model2021 stands for the previous version of the English-Russian system which we consider the baseline.

We should also note that the gain from the transformer-big configuration for the Ukrainian-English model is not that large according to the automatic scores and our human evaluation. We think this is because the synthetic translations obtained from the English-Russian data with the Russian-Ukrainian model are ultimately not of perfect quality.

4 Conclusions and future work

In this paper we presented our submissions for the WMT22 Shared General Translation Task. We show good results in all directions we participate. We clearly outperform our baselines in all directions. A detailed analysis of the translations shows us that we lose quality in translation of colloquial speech. We plan to carefully select colloquial data of very high quality and use it for the general-domain language models for dual cross-entropy data selection. We also plan to train a transformer-big Russian-Ukrainian model and rebuild the synthetic translations for the Ukrainian-English model in the future.

References

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged Back-Translation. In *Proceedings of the Fourth Conference on Machine Translation*, pages 53–63, Florence, Italy.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of NAACL HLT 2013*, pages 644–648, Atlanta, USA.

Marcin Junczys-Dowmunt. 2018. Dual Conditional Cross-Entropy Filtering of Noisy Parallel Corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast Neural Machine Translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, Alexander M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *Computing Research Repository*, arXiv:1701.02810. Version 2.

Marco Lui and Timothy Baldwin. 2012. langid.py: An Off-the-shelf Language Identification Tool. In *Proceedings of ACL 2012, System Demonstrations*, pages 25–30, Jeju, Republic of Korea.

Alexander Molchanov. 2019. PROMT Systems for WMT 2019 Shared Translation Task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 302–307, Florence, Italy.

Kishore Papineni, Salim Roukos, Todd Ward, and WeiJing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 02)*, pages 311–318, Philadelphia, PA, USA.

Marcis Pinnis, Rihards Krišlauks, Daiga Deksne, and Toms Miks. 2017. Neural Machine Translation for Morphologically Rich Languages with Improved Sub-word Units and Synthetic Data. In *Proceedings of the 20th International Conference of Text, Speech and Dialogue (TSD2017)*, pages 237–245, Prague, Czechia.

Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón and Sergio Ortiz Rojas. 2020. Bifixer and Bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298, Lisboa, Portugal.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, Berlin, Germany.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. 2016b. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 35–40, Berlin, Germany.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA.

eTranslation's Submissions to the WMT22 General Machine Translation Task

Csaba Oravecz* **Katina Bontcheva**† **David Kolovratník**†
Bogomil Kovachev* **Christopher Scott***
DG Translation – DG CNECT, European Commission
*firstname.lastname@ec.europa.eu
†firstname.lastname@ext.ec.europa.eu

Abstract

The paper describes the 3 NMT models submitted by the eTranslation team to the WMT22 general machine translation shared task. In the WMT news task last year, multilingual systems with deep and complex architectures utilizing immense amounts of data and resources were dominant. This year with the task extended to cover less domain specific text we expected even more dominance of such systems. In the hope to produce competitive (constrained) systems despite our limited resources, this time we selected only medium resource language pairs, which are serviced in the European Commission's eTranslation system. We took the approach of exploring less resource intensive strategies focusing on data selection and data filtering to improve the performance of baseline systems. With our submitted systems our approach scored competitively according to the automatic rankings in the constrained category, except for the En→Ru model where our submission was only a baseline reference model developed as a by-product of the multilingual setup we built focusing primarily on the En→Uk language pair.

1 Introduction

The eTranslation team is responsible for the development of machine translation systems providing the translation services of the European Commission's eTranslation project[1]. This is a building block of the Connecting Europe Facility (CEF), with the aim of supporting European and national public administrations' information exchange across language barriers in the EU. The project is described in more details in Oravecz et al. (2019).

During the previous years the team's participation in the WMT shared tasks allowed us to explore state-of-the-art methods to develop high quality machine translation systems. However, due to strict resource constraints, these systems do not normally carry over to production environments and there has been a continuous search for the right balance between the use of resources in production environments and the best performing but more complex architectures.

With the news translation shared task extended to being a general MT task the need for more robustness, coverage and consequently more complexity and resources has further increased. We expect a strong competition in these areas, where teams with modest resources might have some inherent disadvantages. Therefore, in this year's experiments we did not consider high resource language pairs (specifically English → German, our constant submission in previous years) and opted for the medium resource French → German and English → Ukrainian language directions. The latter system originated from a multilingual setup including Russian data, so we built and submitted a baseline English → Russian model as well.

2 Data Preparation

In this section we briefly describe the base data sets, the general selection and filtering methods we applied to prepare these initial data sets used to train the first baseline models. Further data selection and augmentation methods to improve the quality of baseline models are described in Section 3.1. We only used the provided parallel and monolingual data, so our submissions all fall into the constrained category.

2.1 Base Data Selection and Filtering

As a general clean-up, we performed the following filtering steps on the parallel data[2]:

[1] https://language-tools.ec.europa.eu

[2] In some subcorpora, only a subset (not necessarily the same) of these steps was applied, depending on the data set. No filtering was used for the dev sets.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 346 - 351
December 7-8, 2022 ©2022 Association for Computational Linguistics

Data set	Fr→De	En→Uk, Ru	En→Uk	En→Ru
Europarl v10	1.79M	–	–	–
Common Crawl	0.42M	0.78M	–	0.78M
News Commentary v16	0.29M	0.34M	–	0.34M
Tilde Model Corpus	4.24M	9.00k	1.00k	8.00k
Dev sets	0.03M	–	–	–
Wiki Titles v3	0.99M	0.70M		0.70M
ParaCrawl	5.64M	12.9M	7.60M	5.30M
OPUS	–	22.9M	22.9M	–
WikiMatrix	1.99M	5.28M	1.50M	3.78M
Yandex	–	1.00M	–	1.00M
UN Parallel	–	9.19M	–	9.19M
Total:	15.39M	53.1M	32.0M	21.1M

Table 1: Number of segments in the filtered parallel data used for baseline bilingual and multilingual models.

- language identification with FastText[3] (Joulin et al., 2016),
- segment deduplication,
- deletion of segments where source/target token ratio exceeds 1:3 (or 3:1),
- deletion of segments longer than 100-150 tokens (depending on language pair),
- exclusion of segments where the ratio between the number of characters and the number of words was below 1.5 or above 40,
- exclusion of segments without a minimum number of alphabetic characters (2),
- exclusion of segments with tokens longer than 40 characters,
- exclusion of segments where the length difference between source and target in the number of tokens was higher than 8,
- removal of segments where source side contained specific noise patterns (in Fr→De ParaCrawl).

These filtering steps led to an average reduction of about 15-20% of the training data with the number of segments as shown in Table 1. For Fr→De, after some manual inspection of the raw WikiMatrix and ParaCrawl data, we decided to experiment with some further clean-up on these data sets, using dual conditional cross-entropy filtering (Junczys-Dowmunt, 2018), where we built the scoring models from a subset of filtered parallel data (7.6M segments) by excluding ParaCrawl and WikiMatrix. We then built models by deleting the worst scoring 5 and 10 % of the two data sets but none of these models was better then the baseline system, so we did not use this filtering in the submission setups. In En→Uk, we experimented with language model based filtering, where we built the language model from the Leipzig corpora and fine tuned the baseline model on the filtered data set, however, it gave no improvement, so this step was not used in the submission systems either.

2.1.1 Monolingual data

In the Fr→De models, where we used back-translation (Sennrich et al., 2016) to improve baseline performance we utilized monolingual data from the various corpora provided. The data was filtered with the same rules (where applicable) as the parallel data (see Section 2.1). Table 2 provides a summary. For the other systems, we didn't use back-translated data in the submissions[4], only the original parallel data sets.

2.1.2 Development and test data

For Fr→De, since the task had been extended from news translation to general MT, where test data was expected from "news, e-commerce, social, and conversational" text, we opted to use a custom built

[3] https://fasttext.cc/docs/en/language-identification.html

[4] See Section 3.1 for experiments with monolingual Ukrainian news. The other monolingual Ukrainian data sets that could have been used for back-translation came too late for us to be able to reschedule the trainings.

Data set	Fr→De
Europarl v10	2.08M
Leipzig mixed	0.99M
Leipzig web	0.99M
News Commentary v16	0.43M
News Crawl 2021	25.0M
Total:	29.29M

Table 2: Number of segments in the filtered monolingual data used for back-translation.

test set for development rather than some previous dev set from the news domain. We extracted a 10k random subset from the filtered original parallel data and manually selected 2k segments for test and validation each. In the manual selection we tried our best to keep segments most representative of the expected domains. These segments were then obviously removed from the training data.

For En→Uk, the validation data was extended with 2k segment pairs randomly extracted from the filtered original parallel data. In addition to the Flores test set, we used 2 development test sets: 10k segment pairs extracted at random from OPUS, and 5k segment pairs extracted from ParaCrawl.

For En→Ru, we extended the validation data again with 2k segment pairs extracted at random from the 2012–2020 dev sets. Beside the Flores test set, we used 2 additional test sets: a 5k random extraction from the parallel data and the provided 2021 news test set. In the latter two language pairs we did not apply manual selection, we considered the test sets already representative enough for the task.

2.2 Pre- and Postprocessing

As in our previous years' systems, we applied the simplest possible workflow without the standard pre- and postprocessing steps of truecasing, or (de)tokenization, and simply used SentencePiece (Kudo, 2018), which allows raw text input/output within the Marian toolkit (Junczys-Dowmunt et al., 2018)[5] in the experiments. In the submission hypotheses, some simple normalization steps were applied in post-processing, similarly to previous years.

3 Trainings

In all experiments we used Marian, as the core tool of our standard NMT framework in the eTranslation service. Trainings were run as multi-GPU setups on 4 NVIDIA V100 GPUs with 16GB RAM, typically for about 30 epochs. In general, except for the first baseline setups, we built only big transformer models, this year even for back-translation, in the hope of getting better quality output for the higher resource consumption. The development scenario was straightforward without much room for experimenting with different parameters or setups due to limited resource availability: for Fr→De, a single set of 4 member ensembles from big transformers, while in En→Uk and En→Ru, a multilingual model at the first stage, fine tuned on the specific languages at the second stage, with 4 (Uk) and 3 (Ru) models in an ensemble as our submission systems for these two language pairs. The parameter settings did not change from last year's setup: for most of the hyperparameters we used the default settings in the baseline models for the base transformer architecture in Marian[6] with dynamic batching and tying all embeddings. In Fr→De, trainings were stopped if sentence-wise normalized cross-entropy on the validation set did not improve in 5 consecutive validation steps. The multilingual systems were stopped after about 40 epochs, and then fine tuned for each target direction until they were stopped to meet the submission deadline.

In the big transformer setups, we also followed standard settings for Marian, i.e. we doubled the filter size and the number of heads, decreased the learning rate from 0.0003 to 0.0002 and halved the update value for `-lr-warmup` and `-lr-decay-inv-sqrt`.

Following common ranges of subword vocabulary sizes, we set a 32k joint SentencePiece vocabulary for all language pairs. SentencePiece models were trained from 10M random segments.

3.1 Synthetic Data

In Fr→De, we back-translated the monolingual data described in Section 2.1.1 with a single big transformer trained from all available original parallel data. The resulting synthetic data set was filtered (where applicable) with the same techniques as the original parallel data. To train the submission

[5] We kept the default settings for Marian's built-in SentencePiece: unigram model, built-in normalization and no subword regularization.

[6] See eg. `https://github.com/marian-nmt/marian-examples/tree/master/transformer`.

		Test sets					
		Dev			2022		
System	Data	COMET	ChrF	BLEU	COMET	ChrF	BLEU
M1: Bilingual baseline	32.0M	69.9	63.2	40.3	47.4	52.9	24.4
M2: Multilingual En→{Uk,Ru}	53.1M	68.4	62.3	39.2	46.7	52.5	24.2
M3: M2 fine-tuned on En→Uk	53.1M/32.0M	71.2	63.7	40.9	50.1	53.3	24.5
M4: $M2^{bigTr}$	53.1M	73.0	64.2	41.5	53.3	54.3	25.6
M5: M4 fine-tuned on $En\rightarrow Uk^{bigTr}$	53.1M/32.0M	74.2	65.0	42.7	52.5	54.4	25.8
M6: 4 x M5 $ens._{subm}^{bigTr}$	53.1M/32.0M	75.0	65.3	43.2	**54.5**	**54.8**	**26.2**

Table 3: Results for En→Uk models. The *Dev* column displays the global scores for all dev sets concatenated.

ready systems we upsampled the (filtered) baseline original parallel (OP) data set to a 1:1 ratio with the BT data (Ng et al., 2019; Junczys-Dowmunt, 2019). This setup was a one shot configuration, we lacked the resources to experiment with other OP-BT combinations. As in previous years, we used tagged back-translation (Caswell et al., 2019) in our workflows.

In En→Uk, back-translation of a 2.4 M subset of monolingual news data with a reverse engine trained from original parallel data did not yield any improvement over the baseline so it was not used in the submission systems.

3.2 Continued Trainings

For Fr→De, in the first phase of the trainings we used all available OP data together with the back-translated synthetic data set. As a second phase after model convergence, we continued the training for 3 additional epochs[7] only on the OP data set.

In the multilingual setup, the first phase of the trainings utilized all available OP data for En→Ru and En→Uk[8]. These trainings were stopped after about 40 epochs and continued only on the respective target data. In both phases the source language data was prefixed with the target language code. All continued trainings were stopped before the submission deadline.

4 Results

We submitted a constrained system for each of the 3 language pairs. We provide COMET (Rei et al., 2020) (with the default model `wmt20-comet-da`), ChrF (Popović, 2017) and BLEU (Papineni et al., 2002) evaluation scores for models at important stages in the development, which reflect how the performance of the models changed as we experimented with the various configurations.[9]

4.1 English→Ukrainian

Table 3 gives a summary of the of the En→Uk experiments. The baseline model (M1) was trained on the filtered original parallel (OP) data using the base transformer architecture. We did not primarily go for a system with synthetic data since the usable monolingual Uk data was small in size (2.6M after filtering) and we didn't expect substantial improvement. Instead, we decided to experiment with multilingual systems. The next model (M2) was a multilingual En→{Uk,Ru} system trained only on filtered OP data (En→Uk, En→Ru), again as a base transformer. The target language was indicated in a token that was prefixed to the source language segments. The slight drop of the scores compared to M1 is not unexpected in multilingual NMT systems when using the same architecture as the bilingual model (Wang et al., 2020). In the next step we used the model of M2 that scored best on the En→Uk development test sets and fine-tuned on En→Uk data until convergence (early-stopping set to 20 stalls). This fine-tuned model was better than the bilingual baseline (M1) and the multilingual M2. The next step (M4) was to train M2 with big transformer architecture. This model was significantly better than all 3 previous models. M5 was an M4 model fine-tuned on En→Uk data, while M6 (our submission model) was a 4 member ensemble built from M5 models. Both M5 and M6 yielded some

[7] We experimented with different number of epochs, until we saw a steady improvement on the test set.

[8] Without EU-Acts, which came too late.

[9] sacreBLEU signatures:

```
chrF2|nrefs:1|case:mixed|eff:yes|nc:6|nw:0|
space:no|version:2.1.0
BLEU|nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|
version:2.1.0
```

		Test sets					
		Dev			2022		
System	Data	COMET	ChrF	BLEU	COMET	ChrF	BLEU
M1: Bilingual baseline	21.1M	51.2	57.2	31.8	48.3	53.4	27.0
M2: Multilingual En→{Uk,Ru}	53.1M	50.3	56.9	31.1	47.3	53.1	26.7
M3: M2^{bigTr}	53.1M	57.8	59.5	34.1	56.2	55.4	29.2
M4: M3 fine-tuned on En→Ru^{bigTr}	53.1M/21.1M	59.6	59.9	34.8	56.1	55.3	29.1
M5: 3 x M4 ens.$^{bigTr}_{subm}$	53.1M/21.1M	60.3	60.3	35.3	**57.9**	**55.8**	**29.8**

Table 4: Results for En→Ru models. The *Dev* column displays the global scores for all dev sets concatenated.

		Test sets					
		Dev			2022		
System	Data	COMET	ChrF	BLEU	COMET	ChrF	BLEU
M1: Baseline	15.4M	64.6	62.1	32.9	47.2	65.2	41.5
M2: M1+BT^{bigTr}	59.4M	65.1	62.3	33.0	53.1	67.1	44.5
M3: M2 cont.^{bigTr}	59.4M	65.5	62.4	33.1	53.4	67.3	44.9
M4: 4 x M3 ens.$^{bigTr}_{subm}$	59.4M	66.7	62.8	34.0	**55.4**	**68.4**	**46.5**

Table 5: Results for Fr→De models.

improvement in the automatic metrics.

4.2 English→Russian

The main stages of the model development for the En→Ru language pair are presented in Table 4. As we described before, the En→Ru system was not intended to be a competitive submission, and this is reflected in the evaluation scores, which are below the scores of other submissions. The baseline model (M1) was trained on the filtered OP data as a base transformer. The next two models (M2 and M3) are common with En→Uk (M2 and M4) – a multilingual En→{Uk,Ru} systems trained only on filtered OP data as base/big transformers (cf. Section 4.1 above). M4 is the M3 model fine-tuned on En→Ru OP data, while M5 (our submission model) is a 3 member ensemble built from M4 models. The score improvements are similar to En→Uk.

4.3 French→German

Table 5 summarizes the results of the Fr→De experiments. The first baseline model (M1) was trained only on the (filtered) original parallel (OP) data with the base transformer architecture. The next model (M2) switched to the big transformer setup and used the back-translated (BT) data with the OP data upsampled (see Section 3.1). Despite the significant increase of the training data size, the effect on the scores on our development set was moderate, however, on the 2022 test set the increase was substantial. This might suggest that the back-translated data gave better support than the OP data to the 2022 test set as a general test set but was much less effective for our development set (which was perhaps still too restricted to the news domain). In the 3rd model (M3), we continued the training only with the OP data as described in Section 3.2, with a slight increase in the metrics. Our submission model (M4) was a 4 member ensemble built from M3 models, where the 4th model was weighted 10% more than the rest. This configuration yielded the most promising result with a significant increase in the scores suggesting that ensembling might be an efficient strategy for general MT models.[10] Model 4 ended up as the best submission of the constrained category, according to all automatic metrics.

5 Conclusion

We described the submissions of the eTranslation team to the WMT22 general MT shared task on 3 language pairs: French-German, English–

[10] In previous years, ensembling was less efficient in our submitted news specific models.

Ukrainian and English–Russian, the last submission being only a baseline setup for reference, built only as a by-product of the En→Uk system. We selected medium resource language pairs and tried to focus on data selection, filtering and evaluation with custom test sets to be able to produce strong constrained systems even with limited resources. In our two competitive systems, first automatic results seemed to justify this approach.

References

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233, Florence, Italy. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Csaba Oravecz, Katina Bontcheva, Adrien Lardilleux, László Tihanyi, and Andreas Eisele. 2019. eTranslation's submissions to the WMT 2019 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 320–326, Florence, Italy. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020. On negative interference in multilingual models: Findings and a meta-learning treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450, Online. Association for Computational Linguistics.

CUNI Systems for the WMT 22 Czech-Ukrainian Translation Task

Martin Popel **Jindřich Libovický**[*] **Jindřich Helcl**[*]
Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
{popel,libovicky,helcl}@ufal.mff.cuni.cz

Abstract

We present Charles University submissions to the WMT 22 General Translation Shared Task on Czech-Ukrainian and Ukrainian-Czech machine translation. We present two constrained submissions based on block back-translation and tagged back-translation and experiment with rule-based romanization of Ukrainian. Our results show that the romanization only has a minor effect on the translation quality. Further, we describe Charles Translator, a system that was developed in March 2022 as a response to the migration from Ukraine to the Czech Republic. Compared to our constrained systems, it did not use the romanization and used some proprietary data sources.

1 Introduction

How fast can the machine translation (MT) community react to a sudden need of a high-quality MT system which was previously under low demand? This question motivated the new task at the WMT this year, which is Czech-Ukrainian translation.

Both languages belong to the Slavic language family (Czech is western Slavic, Ukrainian is eastern Slavic), and share some lexical and structural characteristics. Unlike Czech, which uses the Latin script, Ukrainian uses its variant of the Cyrillic alphabet.

We submit three systems to the WMT 22 General Translation Shared Task for this language pair in each translation direction. The first system, CUNI-JL-JH, implemented in Marian (Junczys-Dowmunt et al., 2018), uses tagged back-translation and is a result of our experiments with romanization of Ukrainian. Our second system, CUNI-TRANSFORMER, implemented in Tensor2Tensor (Vaswani et al., 2018), uses block back-translation. Finally, we submit an unconstrained system, CHARLES TRANSLATOR, implemented in Tensor2Tensor, which has been developed in spring 2022 as a response to the crisis caused by the Russian invasion of Ukraine and the following migration wave.

2 Constrained WMT Submissions

We submitted two systems in each translation direction that use the same parallel and monolingual data, but different techniques and different toolkits. This section first describes the shared data processing steps and then the specifics of each of the submissions in separate subsections.

2.1 Training Data

We use all parallel data allowed in the constrained task, along with 50 million Czech and 58 million Ukrainian sentences of monolingual data. In the following paragraphs we describe the data cleaning steps when preparing the training data. We further experiment with romanization of the Ukrainian Cyrillic alphabet and with artificial noising of the data.

Parallel data. The data for the constrained translation task consist of OPUS corpora (Tiedemann, 2012) that have a Czech-Ukrainian part, WikiMatrix (Schwenk et al., 2021) and the ELRC EU acts in Ukrainian.[1]

We clean the parallel data using rule-based filtering in the following way:

1. Filter out non-printable and malformed UTF-8 characters.

2. Detect language using FastText (Grave et al., 2018), only keep Czech and Ukrainian sentences on their respective source/target sides.

3. Only keep sentence pairs with character length ratio between 0.67 and 1.5 if longer than 10 characters.

[*] The author order was determined by a coin toss.

[1] https://elrc-share.eu/repository/search/?q=EU+acts+in+Ukrainian

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 352 - 357
December 7-8, 2022 ©2022 Association for Computational Linguistics

Source	Original	Filtered
bible-uedin	8 k	8 k
CCMatrix	3,992 k	3,884 k
EUbookshop	2 k	1 k
GNOME	150	81
KDE4	134 k	64 k
MultiCCAligned	1,607 k	1,199 k
MultiParaCrawl	1,773 k	1,606 k
OpenSubtitles	731 k	273 k
QED	161 k	138 k
Tatoeba	3 k	2 k
TED2020	115 k	106 k
Ubuntu	0.2k	0.2k
wikimedia	2 k	2 k
XLEnt	695 k	695 k
WikiMatrix	105 k	99 k
ELRC EU Acts	130 k	108 k
Total	9,457 k	8,186 k

Table 1: Sizes of parallel data sources (number of sentence pairs).

4. Apply hand-crafted regular expressions to filter out the frequent errors, such that the system does not attempt to translate e-mail addresses, currencies, etc. In addition, regular expressions check translations of names of Czech[2] and Ukrainian[3] municipalities downloaded from Wikipedia.

We omit steps 2 and 3 for the XLEnt corpus, which seems to be very clean and consist of short phrases (likely to get misclassified for language).

The sizes of the used parallel data sources before and after cleaning are presented in Table 1.

Monolingual data. The overview of the monolingual data sources is in Table 2. For Czech, we use the Czech monolingual portion of the CzEng 2.0 corpus (Kocmi et al., 2020). For Ukrainian, we used all resources, available for WMT, i.e., the NewsCrawl, the Leipzig Corpora (Biemann et al., 2007), UberText corpus (Khaburska and Tytyk, 2019) and Legal Ukrainian Crawling by ELRC. The Uber corpus and the Ukrainian Legal corpus are distributed tokenized with removed punctuation. We automatically restored the punctuation and detokenized the models using a lightweight Transformer model (Vaswani et al., 2017; Base model with 3 layers, 8k vocabulary) trained on the NewsCrawl corpus.

For Ukrainian, we only keep sentences shorter than 300 characters. For Czech, we keep all sentence lengths from the CzEng corpus (up to 1400 characters). For both languages, we remove non-printable and malformed UTF-8 characters.

Source		Original	Filtered
Czech	CzEng 2.0		50.6 M
Ukrainian	NewsCrawl	2.3 M	2.0 M
	Leipzig Corpora	9.0 M	7.6 M
	UberText Corpus	47.9 M	41.2 M
	ELRC Legal	7.6 M	7.2 M
	Total	66.8 M	58.1 M

Table 2: Monolingual data sizes in number of sentences before and after filtering.

Romanization. We develop a reversible romanization than transcribes between the Ukrainian and Czech alphabets. For example, Зараз у нас є 4-місячні миші is transcribed to *Zaraz u nas je 4-misjačni myši*. This way the model can better exploit the lexical similarities between the two languages (e.g. миші should be translated to Czech as *myši*), while keeping all the necessary information to reconstruct the original Cyrillic text. Note that the transcription of Cyrillic changes when changing the target language, reflecting the phonology of that language (e.g. ш transcribes to *sh* in English, but *š* in Czech). We introduce special tags for words and characters that are written in Latin script found in Cyrillic text. The romanization is specifically designed for Ukrainian (e.g. и transcribes to *y*, not *i* as would be the case in Russian), so its reversibility occasionally fails for Russian names.

Artificial noise. We apply synthetic noise on the source side that should simulate the most frequent deviations from the standard orthography (missing capitalization, lower- or upper-casing parts of the sentences, missing or additional punctuation).

All scripts for training data processing are available at `https://github.com/ufal/uk-cs-data-scripts`. We use Flores 101 (Goyal et al., 2022) development set for validation.

2.2 Tagged-back-translation-based System (CUNI-JL-JH)

The CUNI-JL-JH submission is a constrained system and uses the data described in the paragraphs above. We train the system in 3 iterations of tagged back-translation (Caswell et al., 2019) with greedy decoding. Each iteration, we filter the back-translated data using Dual Cross-Entropy filtering (Junczys-Dowmunt, 2018) when keeping

[2] https://uk.wikipedia.org/wiki/Міста_Чехії

[3] https://cs.wikipedia.org/wiki/Seznam_měst_na_Ukrajině

$40,930,735$ synthetic sentences, i.e., 5× the size of clean authentic parallel data.

The first two back-translation iterations were done with the Cyrillic script on the Ukrainian side. In the final back-translation iteration, we performed romanization and noising of the source side. We train three models with random initialization and submit the ensemble.

For all iterations, we used a Transformer Big model with tied embeddings and a shared SentencePiece vocabulary size of 32k (fitted on 5M randomly sampled sentences; with sampling at the training time, α=0.1; Kudo and Richardson, 2018). We set the learning rate to 0.0003 and use $8,000$ warm-up steps. We initialize the models randomly in each back-translation iteration.

For validation, we use greedy decoding. At test time, we decode with beam search with beam size of 4 and length normalization of 1.0 (estimated on validation data).

The system is implemented using Marian (Junczys-Dowmunt et al., 2018).

Negative results. We experimented with Dual-Cross-Entropy filtering (Junczys-Dowmunt, 2018) for parallel data selection and came to inconclusive results. Therefore, we used all parallel data after rule-based filtering.[4]

Additionally, we experimented with MASS-style (Song et al., 2019) pre-training using monolingual data only and continue with training on parallel data. We were not able to find a hyper-parameter setting where the pre-trained model would outperform the models trained from random initialization. Therefore, we only use model trained from random initialization.

2.3 Block back-translation System (CUNI-Transformer)

The CUNI-Transformer submission is also constrained, trained on the same data as CUNI-JL-JH. The system was trained in the same way as the sentence-level English-Czech CUNI-Transformer systems submitted to previous years of WMT shared tasks (Popel, 2018, 2020; Gebauer et al., 2021). It uses Block back-translation (BlockBT) (Popel et al., 2020), where blocks of authentic (human-translated parallel) and synthetic (back-translated) training data are not shuffled together, but checkpoint averaging is used to find the optimal ratio of checkpoints from the authentic and synthetic blocks (usually 5:3). The uk→cs system was trained with a non-iterated BlockBT (i.e. cs-mono data was translated with an authentic-only trained baseline). The cs→uk was trained with two iterations of BlockBT (i.e. the uk-mono data was translated with the above mentioned uk→cs non-iterated BlockBT system). We had not enough time to train more iterations and apply noised training and romanization. The system was implemented using Tensor2Tensor (Vaswani et al., 2018).

Inline casing. We experimented with Inline casing (InCa) pre-processing in the cs→uk direction. The main idea is to lowercase all training data and insert special tags <titlecase> and <all-uppercase> before words in the respective case, so that the original casing can be reconstructed (with the exception of words like *McDonald* or *iPhone*, which use different casing patterns than all-lowercase, all-uppercase and titlecase). We improved this approach by remembering the most frequent casing variant of each (lowercased) word in the training data. The most frequent variant does not need to be prefixed with any tag, which makes the length of training sequences shorter. We also introduced a third tag <all-lowercase> for encoding all-lowercased words whose most frequent variant is different. For example, if the InCa vocabulary includes only two items: *iPhone* and *GB*, sentence *My iPhone 64GB and iPod 64 GB or 32 gb* will be encoded as <titlecase> *my iphone* <all-uppercase> *64gb and iPod 64 gb or 32 <all-lowercase> gb*. Note that *iPod* was kept in the original case because it was not included in the InCa vocabulary and it does not match any of the three "regular" casing patterns. We applied InCa on both the source and target side and experimented with training the InCa vocabulary on the authentic data only or on authentic plus synthetic (monolingual backtranslated).

Inline casing showed promising results in preliminary experiments (without backtranslation), especially when combined with romanization and artificial noise in training. Unfortunately, we had not enough time to train the backtranslated model long enough, so we submitted it only as a contrastive run and plan to explore it more in future.

[4] Note that we use Dual-Cross-Entropy for filtering the monolingual data, as described in the first paragraph of this section, but we have not done any experiments with keeping all the monolingual data.

Model	cs→uk	uk→cs
Authentic only	20.91	22.95
BT iteration 1	21.69	23.70
BT iteration 2	21.87	23.98
BT iteration 3 (seed 1)	21.53	23.76

Table 3: Validation BLEU scores for the first two iterations of BT for the tagged BT systems.

3 Charles Translator for Ukraine

Charles Translator for Ukraine is a free Czech-Ukrainian online translation service available for public at `https://translator.cuni.cz` and as an Android app. It was developed at Charles University in March 2022 to help refugees from Ukraine by narrowing the communication gap between them and other people in Czechia. Similarly to CUNI-Transformer, it is based on Transformer and iterated Block back-translation (Popel et al., 2020). The training used source-side artificial noising, but no romanization and no inline casing. It was trained on most (but not all) of the training data provided by WMT plus about one million uk-cs sentences from the InterCorp v14 corpus (Čermák and Rosen, 2012; Kotsyba, 2022), so this submission is unconstrained.

4 Results

In this section, we report BLEU scores on the Flores 101 development set that we used to make our decisions about the system development and the final automatic scores. Note that the validation set is very different from the test set. The validation set consists of clean and rather complicated sentences from Wikipedia articles, whereas the WMT 22 test set is noisy user-generated text from the logs of the production deployment of Charles Translator.[5]

Tagged BT systems. Table 3 shows validation BLEU scores from the first three iterations of back-translation. The second and third iteration did not bring substantial improvements, so we decided not to further iterate.

Table 4 shows validation BLEU scores from the last (third) BT iteration – three independently trained systems and their ensembles, and the Cyrillic and romanized versions of the data. In general, ensembling only brings a small improvement. Romanization does not bring a significant difference compared to using the Cyrillic script. In the Czech-to-Ukrainian direction, the best system was the ensemble of the romanized systems. However, in the Ukrainian-to-Czech direction, the best system was one of the Cyrillic systems that used accidentally 3 times higher batch size than the remaining ones. This result suggests that the batch size has a much stronger effect than most of the techniques that we experimented with and that we might have reached better results if we opted for higher batch size.

Model		cs→uk	uk→cs
Cyrillic	Seed 1	21.53	23.76
	Seed 2	22.28	**25.10**
	Seed 3	21.96	24.39
	Ensemble	22.45	24.86
Romanized	Seed 1	21.42	23.99
	Seed 2	21.76	23.91
	Seed 3	22.37	24.18
	Ensemble	**22.62**	24.22

Table 4: Validation BLEU scores for the last (i.e., the third) iteration of BT comparing romanized and original script.

Results on WMT test. Automatic evaluation on the WMT22 test set is presented in Table 5. Both the constrained systems and Charles Translator show comparable results. The tagged BT system reaches a slightly higher COMET score than the Block BT system, however, Czech-Ukrainian was not in the training data of the COMET score, which make the score unreliable for this particular language pair. For Czech-to-Ukrainian, Charles Translator reaches a slightly higher COMET score and slightly lower BLEU and chrF scores than both the constrained systems, but we do not consider such small differences of automatic metrics relevant.

5 Conclusions

We presented Charles University submissions to the WMT 22 General Translation Shared Task on Czech-Ukrainian and Ukrainian-Czech machine translation. We present two constrained submissions based on block back-translation and tagged back-translation and experiment with rule-based romanization of Ukrainian. Further, we describe Charles Translator, a system that was developed in March 2022 as a response to the migration from Ukraine to the Czech Republic. Compared to our constrained systems, it did not use the romanization

[5] The test set only contains sentences from users who provided their consent for this usage and the sentences were pseudonymized.

	cs→uk			uk→cs		
System	BLEU	chrF	COMET	BLEU	chrF	COMET
Best constrained (HuaweiTSC/AMU)	36.0	62.6	0.994	37.0	60.7	1.048
CUNI-Transformer	35.0	61.6	0.873	35.8	59.0	0.885
CUNI-JL-JL	34.8	61.6	0.900	35.1	58.7	0.890
Best unconstrained (Lan-Bridge/Online-B)	38.1	64.0	0.942	36.5	60.4	0.965
Charles Translator	34.3	61.5	0.908	35.9	59.0	0.901

Table 5: Final automatic results on the WTM22 test data compared to the best overall score achieved in each metric.

and used some proprietary data sources.

Our results show that the romanization only has a minor effect on the translation quality, compared to machine-learning aspects that affect translation quality. Block back-translation seems to deliver slightly better results that tagged back-translation, however the differences are only small.

Acknowledgements

This work has been supported by the Ministry of Education, Youth and Sports of the Czech Republic, Project No. LM2018101 LINDAT/CLARIAH-CZ, by the Czech Science Foundation (GACR) grant 20-16819X (LUSyD), and by the European Commission via its Horizon 2020 research and innovation programme no. 870930 (WELCOME), Horizon Europe Innovation programme no. 101070350 (HPLT).

References

Chris Biemann, Gerhard Heyer, Uwe Quasthoff, and Matthias Richter. 2007. The leipzig corpora collection-monolingual corpora of standard size. *Proceedings of Corpus Linguistic*, 2007.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

František Čermák and Alexandr Rosen. 2012. The case of InterCorp, a multilingual parallel corpus. *International Journal of Corpus Linguistics*, 17(3):411–427.

Petr Gebauer, Ondřej Bojar, Vojtěch Švandelík, and Martin Popel. 2021. CUNI systems in WMT21: Revisiting backtranslation techniques for English-Czech NMT. In *Proceedings of the Sixth Conference on Machine Translation*, pages 123–129, Online. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Anastasiia Khaburska and Igor Tytyk. 2019. Toward language modeling for the ukrainian. *Advances in Data Mining, Machine Learning, and Computer Vision. Proceedings*, pages 71–80.

Tom Kocmi, Martin Popel, and Ondrej Bojar. 2020. Announcing CzEng 2.0 parallel corpus with over 2 gigawords. *CoRR*, abs/2007.03006.

Natalia Kotsyba. 2022. Ukrainian-Czech part of InterCorp v14. `https://intercorp.korpus.cz`.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Martin Popel. 2018. CUNI transformer neural MT system for WMT18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 482–487, Belgium, Brussels. Association for Computational Linguistics.

Martin Popel. 2020. CUNI English-Czech and English-Polish systems in WMT20: Robust document-level training. In *Proceedings of the Fifth Conference on Machine Translation*, pages 269–273, Online. Association for Computational Linguistics.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(4381):1–15.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2Tensor for neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 193–199, Boston, MA. Association for Machine Translation in the Americas.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 6000–6010, Long Beach, CA, USA. Curran Associates, Inc.

The ARC-NKUA submission for the English-Ukrainian General Machine Translation Shared Task at WMT22

Dimitrios Roussis[1,2] **and Vassilis Papavassiliou**[1]
[1]Athena Research Center, Athens, Greece
[2]National and Kapodistrian University of Athens, Greece
{dimitris.roussis, vpapa}@athenarc.gr

Abstract

In what follows, we provide an overview of the ARC-NKUA ("Athena" Research Center - National and Kapodistrian University of Athens) submission to the WMT22 General Machine Translation shared task for the EN-UK (English to Ukrainian) and UK-EN (Ukrainian to English) translation directions. We describe how we constructed two Neural Machine Translation systems by training Transformer models (Vaswani et al., 2017), as well as our experiments involving: (a) ensemble decoding, (b) selected fine-tuning with a subset of the training data, (c) data augmentation with back-translated monolingual data, and (d) post-processing of the translation outputs. Furthermore, we discuss filtering techniques and the acquisition of additional data used for training the systems.

1 Introduction

Neural Machine Translation (NMT) has achieved significant improvements in translation quality in recent years, especially concerning high-resource language pairs. However, there is a lot of room for research on systems with general translation capabilities, underrepresented domains, low- or medium- resource language pairs, as well as multilingual systems. This year, the former news translation shared task widened in scope by introducing new domains, as well as the English-Ukrainian language pair among others.

We participated in the WMT22 General Machine Translation shared task for the unconstrained tracks of the EN-UK (English to Ukrainian) and UK-EN (Ukrainian to English) translation directions. The two submitted NMT systems are based on the Transformer architecture (Vaswani et al., 2017) and our experiments involve various methods and techniques such as data acquisition, filtering and selection, fine-tuning, ensemble decoding, tagged back-translation of English and Ukrainian monolingual sentences and post-processing of the translation outputs.

This paper is structured in the following way: In Section 2, we describe the parallel and monolingual corpora, as well as the acquisition, selection, filtering and pre-processing techniques that were used in our experiments. Section 3 outlines the NMT systems architecture, training parameters and the various experiments on top of our baseline systems. In Section 4, we report and discuss the experimental results of the two translation directions we participated in, while Section 5 concludes and summarizes our work.

2 Datasets

We participated in the unconstrained tracks of this year's general machine translation shared task for the English-Ukrainian and Ukrainian-English translation directions. We made use of most of the datasets given by the organizers: corpora from OPUS[1] (Tiedemann, 2012), ParaCrawl v9[2] and ELRC - EU acts in Ukrainian[3] from the ELRC-SHARE repository. Other parallel resources from this repository that were used in our systems include:

[1] https://opus.nlpl.eu/

[2] https://paracrawl.eu/news/item/17-english-ukrainian-bonus-parallel-corpus

[3] https://elrc-share.eu/repository/eu-acts-in-ukrainian/

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 358 - 365
December 7-8, 2022 ©2022 Association for Computational Linguistics

- Multilingual English, French, Polish to Ukrainian Parallel Corpus (processed)[4]

- Official web-portal of the Parliament of Ukraine, primary legislation[5]

- Official web-portal of the Parliament of Ukraine, Ukrainian laws in EN[6]

- Official web-portal of the Parliament of Ukraine, abstracts of UK laws[7]

- SciPar UK-EN-RU[8] (Roussis et al., 2022a)

- A Bilingual English-Ukrainian Lexicon of Named Entities Extracted from Wikipedia[9]

We also made use of three monolingual datasets given by the organizers: News crawl[10], Leipzig Corpora[11] and Legal Ukrainian Crawling[12] from the ELRC-SHARE repository. After manually inspecting the other given dataset, UberText Corpus[13], we decided not to use it for back-translation (see Section 3.2), as most punctuation is missing. Instead, we make use of monolingual corpora that we acquired (see Section 2.1), as well as the Ukrainian monolingual corpus of WikiMatrix.

2.1 Acquisition of Additional Corpora

In order to acquire additional parallel English-Ukrainian data, we used the ILSP-FC toolkit[14] (Papavassiliou et al., 2013) to crawl candidate parallel documents from websites and the LASER toolkit[15] (Artetxe and Schwenk, 2019) to mine bitexts with the use of its margin-based alignment score, after splitting each document into sentences. It is worth noting that manual inspection was also moderately applied so as to exclude machine translated websites. Additional parallel data acquisition techniques that were used are mentioned in more detail in Roussis et al. (2022a; 2022b). During parallel data acquisition, monolingual sentences in English and Ukrainian were also collected and were later used for back-translation (see Section 3.2).

The aforementioned techniques were used to compile the first five bulleted corpora listed in section 2, as well as EU acts in Ukrainian which was given by the organizers. Nevertheless, we attempted to enrich the acquired data by also targeting approximately 300 websites to extract EN-UK parallel sentences and more than 2,000 websites to extract monolingual UK sentences. This process resulted in ~2M additional EN-UK sentence pairs and ~31.9M monolingual UK sentences.

2.2 Parallel Corpus Filtering

The following filtering methods are used on all of the parallel data (including the subset that we selected for fine-tuning, as well as the synthetic data) after punctuation normalization and tokenization with the Moses toolkit[16] (Koehn et al., 2007):

- Sentence pairs with identical source and target sides are removed (Papavassiliou et al., 2018; Pinnis, 2018).

- Duplicate sentence pairs are removed, based on either source or target side; i.e. no English or Ukrainian sentence (after being lowercased and having its digits removed) appears more than once in the training set.

[4] https://elrc-share.eu/repository/multilingual-english-french-polish-to-ukrainian-parallel-corpus-processed/
[5] https://elrc-share.eu/repository/official-web-portal-of-the-parliament-of-ukraine-primary-legislation/
[6] https://elrc-share.eu/repository/official-web-portal-of-the-parliament-of-ukraine-ukrainian-laws-in-en/
[7] https://elrc-share.eu/repository/official-web-portal-of-the-parliament-of-ukraine-abstracts-of-uk-laws/
[8] https://elrc-share.eu/repository/scipar-uk-en-ru/
[9] https://elrc-share.eu/repository/a-bilingual-english-ukrainian-lexicon-of-named-entities-extracted-from-wikipedia/
[10] http://data.statmt.org/news-crawl
[11] https://wortschatz.uni-leipzig.de/en/download/ukr/
[12] https://elrc-share.eu/repository/legal-ukrainian-crawling/
[13] https://lang.org.ua/en/corpora/#anchor5
[14] http://nlp.ilsp.gr/redmine/projects/ilsp-fc/
[15] https://github.com/facebookresearch/LASER/
[16] https://github.com/moses-smt/mosesdecoder/

- Sentence pairs in which either side consists of more than 50% non-alphabetic characters are removed (Rikters, 2018).

- Sentence pairs in which the length ratio in terms of digit characters is over 2:1 (or below 1:2) are removed.

- Sentence pairs in which either the source or target sentence contains more than 250 tokens or more than 1000 characters are removed.

- Sentence pairs in which the token ratio between the longest and the shortest sentence is higher than 2 are removed.

- Sentence pairs in which either sentence contains letters not in the range of Unicode character sets relevant to Latin and Cyrillic scripts are removed (Papavassiliou et al., 2018).

- The repeating token filter[17] from Rikters (2018) was used to remove sentence pairs originating from machine-translated content.

- Language identification with fastText[18] (Joulin et al., 2017) is used to remove sentence pairs with different languages than expected.

In Table 1, we report the number of the raw (unfiltered) English-Ukrainian sentence pairs (57.7M), the number actually used for training the baseline systems after filtering (19M), and the selected subset used for fine-tuning (10.2M). Additionally, we list the number of filtered synthetic parallel sentences generated from English monolingual sentences with the EN-UK system (60M) and from Ukrainian monolingual sentences with the UK-EN system (54.5M).

2.3 Data Selection

As we will describe in more detail in Section 3.4, we also experimented with fine-tuning the NMT systems (see Section 3.3). In particular, after training a system we continue its training with a subset of the parallel data which has been selected according to some stricter criteria. LASER-based corpus filtering has been shown to have promising results (Chaudhary et al., 2019), it has already been computed for many of the used datasets and we believe that it may prove especially useful in counteracting possible quality degradation in NMT systems trained with additional back-translated data (Tran et al., 2021).

Type of data	Sentence pairs
Raw EN-UK parallel	57,727,556
Filtered EN-UK parallel	19,023,045
Filtered EN-UK parallel selected for fine-tuning	10,203,198
Back-translated from monolingual EN	60,055,592
Back-translated from monolingual UK	54,517,999

Table 1: Number of used EN-UK sentence pairs

To this end, we decided to select an appropriate subset of the training data with the utilization of the alignment score given by the LASER toolkit. For this reason, the LASER scores of the parallel sentences of the available corpora were examined and the following data selection strategy was adopted:

- A LASER score threshold of 1.1 was set for sentence pairs originating from the CCMatrix, CCAligned and ParaCrawl corpora. These three datasets contain a total of 42.1M raw sentence pairs and have been collected from the web.

- A LASER score threshold of 1.06 was set for sentence pairs originating from the WikiMatrix corpus as well as for those that we acquired (see Section 2.1).

2.4 Pre-Processing and Vocabulary

As mentioned in section 2.2, the Moses toolkit is used to normalize the punctuation and tokenize the datasets. Additionally, in order to handle casing, we use the "inline casing" technique (Bérard et al., 2019; Etchegoyhen and Ugarte, 2020; Molchanov, 2020) which uses specific tags to denote uppercase (<UC>), title case (<TC>) or mixed case (<MC>) words. Depending on the tags which the decoder has generated, the output sentences are re-cased

[17] https://github.com/M4t1ss/parallel-corpora-tools/blob/master/parallel/repeating-tokens.php

[18] https://fasttext.cc/docs/en/language-identification.html

during post-processing. Inline casing has been shown as the optimal approach in handling casing (Etchegoyhen and Ugarte, 2020).

After the application of the filtering pipeline, as well as the addition of tags (from inline casing or tagged back-translation) and NFC Unicode normalization, a separate BPE tokenizer with 18k merge operations is trained independently for English and Ukrainian with SubwordNMT [19] (Sennrich et al., 2016a) and BPE-dropout with probability of 0.1 is applied on the source sentences for each translation direction (Provilkov et al., 2020).

3 System Overview

Both submitted systems follow the Transformer architecture (Vaswani et al., 2017) and were trained using two RTX 2080 Ti GPUs with the utilization of the Fairseq toolkit (Ott et al., 2019). In the subsections that follow, we describe the training process of both NMT systems, as well as the techniques that we experimented with in order to improve translation quality.

3.1 Model Architecture and Training

The "big Transformer" architecture (Vaswani et al., 2017) is used as our NMT model, although we made use of 8 encoder layers instead of 6, as increasing the number of encoder layers has been shown to improve performance in many scenarios (Subramanian et al., 2021; Wang et al., 2021b). We apply dropout with probability 0.3, activation dropout with probability 0.1 and attention dropout with probability 0.1. The Adam optimizer (Kingma and Ba, 2014) is used with a peak learning rate of 0.0007 after 4,000 warmup steps which then follows inverse square root decay. The models are trained using half precision training (FP16), with 2,800 tokens per batch, while the parameters are updated every 4 batches (Ott et al., 2018). Checkpoints are saved every 20,000 updates and every 10,000 updates when fine-tuning, while the training stops if the BLEU score on the validation set does not improve for 5 checkpoints. Finally, checkpoint averaging 5 was applied to all NMT systems, i.e., we average the parameters of the 5 last checkpoints in order to obtain the final model parameters.

3.2 Tagged Back-Translation

Back-translation (Sennrich et al., 2016b; Edunov et al., 2018) has been proven as an effective data augmentation technique which leverages large amounts of monolingual data and is particularly useful for domain adaptation and low-resource settings (Bérard et al., 2019; Wang et al., 2021a; Wang et al., 2021b). We follow Caswell et al. (2019) in using tagged back-translation, i.e., inserting a <BT> tag in the beginning of each source sentence which has been synthetically generated; a method which is simple and robust.

For each of the two translation directions, the reverse fine-tuned models trained on parallel data are used (with beam size 5) in order to generate the synthetic outputs (see Table 1). When we enrich the training set with back-translated data, we upsample the original parallel data by a factor of 2.

3.3 Selected Fine-Tuning

Fine-tuning is usually used to adapt a NMT model to a specific domain, i.e., to improve its quality on inputs with specific characteristics. Since this year the former news translation shared task changed its focus to more general translation capabilities, there is not a specific domain which we would like our systems adapted to.

Nevertheless, fine-tuning has also been shown to have a corrective effect on systems which exhibit decreased performance after having been trained with large amounts of synthetic data (Tran et al., 2021; Wang et al., 2021a). Thus, after the training of the NMT models ends, we continue to train them using a selected subset of the training set (see Section 2.3), while also halving the dropout probability to 0.15.

3.4 Ensemble Decoding

Ensemble decoding has been shown to have mostly minor effects on performance, although it can improve performance on specific translation directions (Oravecz et al., 2020; Tran et al., 2021; Subramanian et al., 2021; Wang et al., 2021a; Wang et al., 2021b). During inference, the probability distributions over the next token are averaged

[19] https://github.com/rsennrich/subword-nmt

#	System	EN - UK		UK - EN	
		FLORES101	WMT22	FLORES101	WMT22
(1)	Baseline	30.7	24.2	36.4	40.9
(2)	(1) + Selected Fine-Tuning	**31.0**	**24.4**	36.8	41.5
(3)	(1) + Back-Translation	30.5	23.7	37.4	40.9
(4)	(3) + Selected Fine-Tuning	30.8	24.0	37.7	41.7
(5)	Ensemble	30.7	24.0	**37.8**	**41.9**
WMT22	Best + Post-Processing	-	**25.2**	-	**41.9**

Table 2: BLEU scores on FLORES101 and WMT22 test sets for
English to Ukrainian (EN-UK) and Ukrainian to English (UK-EN) systems.

according to the systems used in ensemble decoding.

It is generally better to use ensemble decoding with NMT systems trained with different seeds or different subsets of the training set (Oravecz et al., 2020; Subramanian et al., 2021). Unfortunately, hardware and time constraints did not allow us to follow this approach and thus, we experimented with ensembling 2 or 3 models from the resulting systems mentioned in the paper.

3.5 Post-Processing

In the WMT 2022 test data provided by the organizers, we observed specific peculiarities which were handled by post-processing scripts. In particular, the Ukrainian data used in the evaluation of the Ukrainian-English systems contained emojis which our systems were not able to handle. We used a simple post-processing script on the English outputs to copy emojis from the beginning or the end of the original Ukrainian input sentences. As regards the Ukrainian outputs of the English-Ukrainian systems, we used a script to replace double quotes ("…") with angled quotation marks («…»), as well as to fix anonymous placeholders according to their original style in the English inputs.

4 Results

We perform the evaluation of our systems using the FLORES101 test set (Goyal et al., 2022) and the WMT22 General Machine Translation test set given by the organizers. Scores are reported in terms of the detokenized case-sensitive BLEU score (Papineni et al., 2002) and have been computed with the SacreBLEU toolkit[20] (Post, 2018). In Table 2, we can see the resulting scores from our experiments, as well as the scores of the submitted models.

4.1 English to Ukrainian

The submitted NMT system for English to Ukrainian has been trained only with parallel data, fine-tuned with a subset of them (see Section 3.3) and its outputs have been post-processed (see Section 3.5). In Table 2, we can see that the effect of back-translation is negative for the EN-UK system. Selected fine-tuning exhibited a corrective effective which, nevertheless, was not enough to offset the initial degradation caused by the addition of synthetic data. However, we also obtain a small improvement (+0.2 BLEU on the WMT22 test set) when fine-tuning the baseline system trained only with parallel data. The largest increase in BLEU scores (+0.8) on the WMT22 test set, is observed after the application of post-processing on the outputs of the final system, which has been trained only with parallel data and fine-tuned on a selected subset of them. This increase does not concern the FLORES101 test set, since there are significant differences in the use of quotation marks between the two test sets. Finally, ensemble decoding did not provide any advantage in our experiments.

4.2 Ukrainian to English

As we can see in Table 2, back-translation initially degrades translation quality but, contrary to the results discussed in Section 4.1, ultimately leads to increased performance after fine-tuning with a selected set of the training data. Ensemble decoding usually has a marginal effect on NMT systems and we see a small increase by its use here as well. For this translation direction, we do not observe any significant difference after the application of post-processing, although we

[20] https://github.com/mjpost/sacreBLEU

decided to use it in the final system, since we do not believe it has any negative effects (less than 50 sentences were affected). Thus, the submitted NMT system for Ukrainian to English is based on all the techniques that we experimented with: back-translation (see Section 3.2), selected fine-tuning (see Section 3.3), ensemble decoding (see Section 3.4) and post-processing (see Section 3.5).

5 Conclusion

In this paper, we have presented the ARC-NKUA submission to the WMT22 General Machine Translation shared task for the English to Ukrainian and Ukrainian to English translation directions. The submitted systems follow the Transformer architecture and were determined after experimentation with back-translation, selected fine-tuning, and ensemble decoding. We showed that the corrective effect of fine-tuning with a subset of the training set can ultimately increase the translation quality of a system which has exhibited degradation due to having been exposed to a large number of synthetic data, while it also proved useful for systems trained only with parallel data.

Our systems underperformed in comparison with other submitted systems, according to automatic scores calculated by the organizers[21], although human judgements will be used for official ranking. In the future, we aim at better investigating the effects of acquiring additional parallel and monolingual data, following different filtering, selection and pre-processing strategies, as well as implementing several techniques which have been generally shown to increase translation quality, but hardware and time constraints did not allow us to experiment upon. Possible techniques that could be investigated include reranking, larger NMT model architecture, iterative back-translation, ensembling models trained on different subsets of the training set and exploiting higher-resource similar languages.

6 Acknowledgements

This work has been supported by the European Language Resource Coordination (ELRC), a service contract under the EC's Connecting Europe Facility SMART 2019/1083 programme.

References

Mikel Artetxe, and Holger Schwenk. 2019. Margin-based Parallel Corpus Mining with Multilingual Sentence Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197-3203, Florence, Italy. Association for Computational Linguistics.

Alexandre Bérard, Ioan Calapodescu, and Claude Roux. 2019. Naver Labs Europe's Systems for the WMT19 Machine Translation Robustness Task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 526-532, Florence, Italy. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged Back-Translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53-63, Florence, Italy. Association for Computational Linguistics.

Vishrav Chaudhary, Yuqing Tang, Franscisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-Resource Corpus Filtering using Multilingual Sentence Embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 261-266, Florence, Italy. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding Back-Translation at Scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489-500, Brussels, Belgium. Association for Computational Linguistics.

Thierry Etchegoyhen, and Harritxu G. Ugarte. 2020. To Case or not to case: Evaluating Casing Methods for Neural Machine Translation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3752-3760, Marseille, France. European Language Resources Association.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Franscisco Guzmán, and Angela Fan. 2022. The Flores-101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation. *Transaction of the Association for Computational Linguistics*, 10:522-538.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th*

[21] https://github.com/wmt-conference/wmt22-news-systems

Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 427-431, Valencia, Spain. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Diederik P. Kingma, and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

Alexander Molchanov. 2020. PROMT Systems for WMT 2020 Shared News Translation Task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 248-253, Online. Association for Computational Linguistics.

Csaba Oravecz, Katina Bontcheva, László Tihanyi, David Kolovratnik, Bhavani Bhaskar, Adrien Lardilleux, Szymon Klocek, and Andreas Eisele. 2020. eTranslation's Submissions to the WMT 2020 News Translation Task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 254-261, Online. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1-9, Brussels, Belgium. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48-53, Minneapolis, Minnesota. Association for Computational Linguistics.

Vassilis Papavassiliou, Prokopis Prokopidis, and Gregor Thurmair. 2013. A modular open-source focused crawler for mining monolingual and bilingual corpora from the web. In *Proceedings of the sixth workshop on building and using comparable corpora*, pages 43-51, Sofia, Bulgaria. Association for Computational Linguistics.

Vassilis Papavassiliou, Sokratis Sofianopoulos, Prokopis Prokopidis, and Stelios Piperidis. 2018. The ILSP/ARC submission to the WMT 2018 Parallel Corpus Filtering Shared Task. In *Proceedings of the Third Conference of Machine translation: Shared Task Papers*, pages 928-933, Belgium, Brussels. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Mārcis Pinnis. 2018. Tilde's parallel corpus filtering methods for WMT 2018. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 939-945, Belgium, Brussels. Association for Computational Linguistics.

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186-191, Brussels, Belgium. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-Dropout: Simple and Effective Subword Regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882-1892, Online. Association for Computational Linguistics.

Matīss Rikters. 2018. Impact of Corpora Quality on Neural Machine Translation. In *Human Language Technologies–The Baltic Perspective*, pages 126-133. IOS Press.

Dimitris Roussis, Vassilis Papavassiliou, Prokopis Prokopidis, Stelios Piperidis, and Vassilis Katsouros. 2022a. SciPar: A Collection of Parallel Corpora from Scientific Abstracts. In *Proceedings of the 13th Language Resources and Evaluation Conference*, pages. 2652–2657, Marseille, France. European Language Resources Association (ELRA).

Dimitris Roussis, Vassilis Papavassiliou, Sokratis Sofianopoulos, Prokopis Prokopidis, and Stelios Piperidis. 2022b. Constructing Parallel Corpora from COVID-19 News using MediSys Metadata. In *Proceedings of the 13th Language Resources and Evaluation Conference*, pages 1068-1072, Marseille, France. European Language Resources Association (ELRA).

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), pages 1715-1725, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* pages 86-96, Berlin, Germany. Association for Computational Linguistics.

Sandeep Subramanian, Oleksii Hrinchuk, Virginia Adams, and Oleksii Kuchaiev. 2021. NVIDIA NeMo's Neural Machine Translation Systems for English-German and English-Russian News and Biomedical Tasks at WMT21. In *Proceedings of the Sixth Conference on Machine Translation (WMT)*, pages 197-204, Online. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214-2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. Facebook AI's WMT21 News Translation Task Submission. In *Proceedings of the Sixth Conference on Machine Translation (WMT)*, pages 205-215, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. *arXiv preprint arXiv: 1706.03762.*

Weixuan Wang, Wei Peng, Xupeng Meng, and Qun Liu. 2021a. Huawei AARC's Submissions to the WMT21 Biomedical Translation Task: Domain Adaptation from a Practical Perspective. In *Proceedings of the Sixth Conference on Machine Translation*, pages 868-873, Online. Association for Computational Linguistics.

Xing Wang, Tu Zhaopeng, and Shurning Shi. 2021b. Tencent AI Lab Machine Translation Systems for the WMT21 Biomedical Translation Task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 874-878, Online. Association for Computational Linguistics.

The NiuTrans Machine Translation Systems for WMT22

Weiqiao Shan[1], **Zhiquan Cao**[1], **Yuchen Han**[1], **Siming Wu**[1], **Yimin Hu**[1],
Jie Wang[1], **Yi zhang**[1], **Hang Cao**[1], **Baoyu Hou**[1], **Chenghao Gao**[1], **Xiaowen Liu**[1],
Tong Xiao[1,2], **Anxiang Ma**[1,2] **and Jingbo Zhu**[1,2]
[1]NLP Lab, School of Computer Science and Engineering,
Northeastern University, Shenyang, China
[2]NiuTrans Research, Shenyang, China
shanweiqiao96@gmail.com ,{xiaotong, maanxiang, zhujingbo}@mail.neu.edu.cn

Abstract

This paper describes the NiuTrans neural machine translation systems of the WMT22 General MT constrained task. We participate in four directions, including Chinese→English, English→Croatian, and Livonian↔English. Our models are based on several advanced Transformer variants, e.g., Transformer-ODE, Universal Multiscale Transformer (UMST). The main workflow consists of data filtering, large-scale data augmentation (i.e., iterative back-translation, iterative knowledge distillation), and specific-domain fine-tuning. Moreover, we try several multi-domain methods, such as a multi-domain model structure and a multi-domain data clustering method, to rise to this year's newly proposed multi-domain test set challenge. For low-resource scenarios, we build a multi-language translation model to enhance the performance and try to use the pre-trained language model (mBERT) to initialize the translation model.

1 Introduction

We participate in the WMT22 General MT task, including Chinese→English (ZH→EN), English→Croatian (EN→HR), and Livonian↔English (LIV↔EN) in four directions. All of our systems are built with constrained data sets. We adopt some methods that have been proven to work well in WMT over the past few years (Li et al., 2019; Zhang et al., 2020; Zhou et al., 2021). At the same time, we also adopt some new model structures (Li et al., 2022; Jiang et al., 2020), data clustering (Aharoni and Goldberg, 2020), initialization (Guo et al., 2020), and training methods (Liu et al., 2021), which are described in detail below.

For data preparation and augmentation, since filtering data could hurt the model performance on the general domain machine translation task, we apply several soft data filtering rules to preserve as much data as possible (Zhang et al., 2020; Zhou et al., 2021). To obtain the in-domain data, we use the open-source toolkit XenC (Rousseau, 2013) and specially try a domain clusters method based on the BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) model in the ZH→EN direction. We also use back-translation (Sennrich et al., 2016a) and knowledge distillation (Freitag et al., 2017) iteratively to increase the size of in-domain data, which has been proved effective in recent years (Zhang et al., 2020; Zhou et al., 2021).

For model architectures, our system is built on several Transformer variants, including Transformer-RPR, Transformer-DLCL (Wang et al., 2019), Transformer-ODE (Li et al., 2021), Transformer-UMST (Li et al., 2022), and Transformer-based model with domain mixing (Jiang et al., 2020). We build a wide and deep model based on the pre-norm structure (Wang et al., 2019) and relative position representation(RPR) (Shaw et al., 2018), inspired by the effectiveness of the deep model. Furthermore, we select four single models to build the ensemble model for better performance. Particularly, in the EN↔LIV direction, we build a multilingual machine translation system (Johnson et al., 2017) based on the above models.

For model initialization, training, and decoding strategies, we use nucleus sampling(Top-P) (Holtzman et al., 2020), top-k sampling(Top-K), and beam search as decoding methods in all languages. At the same time, we adopt scheduling sampling (Liu et al., 2021) in ZH→EN direction during fine-tuning. Furthermore, we attempt to initialize the translation model with the pre-trained language model based on lightweight adapter (Guo et al., 2020) in the EN↔LIV direction.

Based on the softer filtering rules and appropriate hypo-parameter settings, we achieve better results on the deep model than last year. In the ZH→EN direction, fine-tuning with the normal training and the scheduling sampling also obtain

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 366 - 374
December 7-8, 2022 ©2022 Association for Computational Linguistics

good results. Furthermore, we use an unsupervised multi-domain data clustering method and some simple domain classification methods. However, we find no significant domain differences in the constrained data. Initializing the translation model with the pre-trained model leads to poor performance in the EN↔LIV direction. It may be due to the sensitivity to the size of the training set.

The rest of the paper is organized as follows: In Section 2, we describe our system in detail, including the data preprocessing and filtering, model structure, back-translation and knowledge distillation, fine-tuning, and post-editing. In Section 3, we introduce our experimental settings and results according to different tasks and give a brief analysis. In Section 4, we summarize our work.

2 System Overview

In Figure 1, we describe the whole process of our system. We use three different colors to represent the different translation tasks. At the data preparation stage, we perform several data processing methods to obtain the training set. Then, we train several models with different structures and use back-translation(BT) and knowledge distillation(KD) iteratively based on ensemble model. Finally, we obtain our final submission based on fine-tuning and post editing.

2.1 Data Preprocessing and Filtering

In the word segmentation stage, we choose different word segmentation methods for the three languages according to the language characteristic. In ZH→EN, we use the `NiuTrans` (Xiao et al., 2012) word segmentation tool for both Chinese and English, which makes it easier for the model to align the words in the bilingual sentence. In EN↔LIV, we use `Reldi-Tokeniser`[1] for each language. In EN→HR, we use `Reldi-Tokeniser` for Croatian and `Niutrans` for English. Further, we apply BPE (Sennrich et al., 2016b) with 32K operations and not shared vocabulary in most language pairs. Specifically, in EN↔LIV, we use five languages, including EN, CS, LIV, ET, and LV, to build a multilingual translation system. We apply BPE with different operations for different languages, as shown in Table 1. Furthermore, we manually construct a dictionary based on `fast_align` (Dyer et al., 2013) to improve word-level alignment.

language	operations
EN	32K
CS	32K
LIV	10K
ET	10K
LV	10K

Table 1: Bpe operations in Livonian↔English

We mainly use the previous filtering method (Zhou et al., 2021). Nevertheless, we adopt softer filtering rules to improve the model performance on the general MT task as follows:

- Filter out sentences that contain long words over 40 characters and sentences that contain over 200 words.
- The word ratio between the source and target sentence must be in the range of [1/3, 3].
- Use Unicode to filter uncommon characters that never appear in previous years' test sets.
- Filter out the sentences which contain HTML tags or duplicated translations.

We use the same filtering rules for monolingual and bilingual data, and based on the filtering rules, we retain more data to do domain filtering further. Based on these filtering rules, we effectively reduce the <UNK> proportion on the previous years' newstest set, while retaining some longer sentences to meet the challenge of the general test set.

2.2 Model Architectures

In recent years, the deep model has been widely proven to be a very effective model structure (Wang et al., 2019; Zhang et al., 2020; Zhou et al., 2021), so we use a variety of deep models, including Transformer-RPR, Transformer-DLCL (Wang et al., 2019), and Transformer-ODE (Li et al., 2021). In addition, we use a new model structure, Transformer-UMST (Li et al., 2022), which uses multi-scale information to enhance the representation ability of model representation. The explicit information of the above model is shown in Table 2.

Transformer-RPR: Compare to Vanilla Transformer, we only increase the number of encoder layers and add RPR into the self-attention at each layer to efficiently consider the relative positions between different representations.

[1] https://github.com/clarinsi/reldi-tokeniser/blob/master/LICENSE

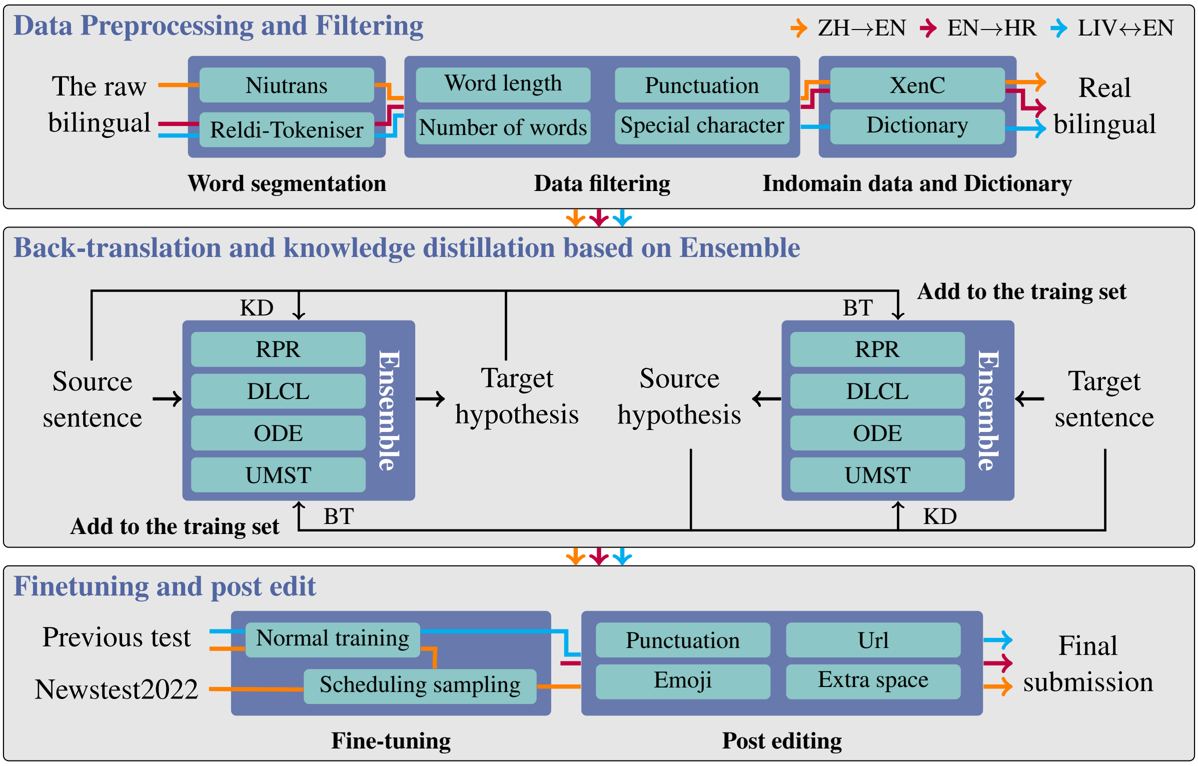


Figure 1: The whole process of our system.

Transformer-DLCL: Build a deeper network with dense inter-layer connections based on the vanilla Transformer, which can increase the information flow at the lower layer.

Transformer-ODE: Based on the relationship between numerical methods of Ordinary Differential Equations(ODEs) and Transformer, A more efficient Transformer calculation method can be obtained by solving ODEs.

Transformer-UMST: Enhance the representation ability of vanilla Transformer by importing sentence-level and word-level information to attention.

2.3 Back-Translation And Knowledge Distillation

Back-translation (Sennrich et al., 2016a) is a popular data augmentation method to improve the performance of machine translation models. We use iterative back-translation(Hoang et al., 2018) based on the in-domain monolingual data to alleviate the domain adaptation problem (Zhang et al., 2020). In addition to use pseudo data directly, we also try Tagged Back-translation (Caswell et al., 2019) in the EN→HR direction. Based on iterative back-translation, we utilize iterative knowledge distillation, which iteratively transforms knowledge (Zhou et al., 2021) from an ensemble model to sub-models based on sequential knowledge distillation (Hinton et al., 2015; Kim and Rush, 2016). We use the following steps for iterative back-translation and knowledge distillation:

1. Select the good quality monolingual data from the source language, filter the data closest to the single domain by XenC toolkit, and obtain bilingual pseudo-data by a forward translation.
2. Filter the data, mix the pseudo data with the training set, and train the back translation model.
3. Search for the best ensemble model combination among all existing models.
4. Use the ensemble model to translate the filtered monolingual data of the target language, and obtain the pseudo data.

We use the `newstest2021` to evaluate our model performance, and repeat steps 1-3(BT) or 2-4(KD) of the above process until the model performance no longer improves[2]. When performing steps 1

[2] It is worth noticing that we do a set of KD after a set of BT, these two methods are combined sequentially

and 4, we use various decoding methods, including beam search, Top-K, and Top-P. In the tasks of ZH→EN, EN→HR, the ratio of raw bilingual data to pseudo data in the training set was about 2 : 1. In the EN↔LIV tasks, the size of pseudo data is much larger than the raw bilingual data.

2.4 Model Ensemble

The ensemble model can significantly improve the translation quality by considering the output of every single model. We search for the model ensemble combined with the highest BLEU score on the `newstest2021` and use the model ensemble repeatedly in knowledge distillation, back-translation, and fine-tuning. This ensures that we can obtain the optimal models at every stage.

2.5 Fine-tuning

A model trained on a large amount of data may not outperform a model trained on a small amount of in-domain data (Zhou et al., 2021). This phenomenon indicates a mismatch between the domain of the training set and the test set, which becomes an obstacle to performance improvement. For a specific domain, we adjust the size of the training datasets that are more focused on a single domain. However, We find it hard to separate bilingual data into multiple domains. In the case that the training set domain is inseparable, fine-tuning by domain is a reasonable way.

Fine-tuning the model with in-domain data is an effective way to alleviate the domain mismatch (Luong and Manning, 2015; Zeng et al., 2021; Tran et al., 2021). In the ZH→EN direction, we split the test set into four domains according to the domain label in the test set and fine-tune the model in the single domain for each of the four domains.

Take the news domain as an example, and fine-tuning process consists of the following three steps:

1. Translate sentences in the news domain to generate pseudo data by the best ensemble model on `newstest2021`.

2. Fine-tune all sub-models in the ensemble model with pseudo-data, `newstest2020`, and `newstest2021`.

3. Based on the data mentioned in the previous step and `test2022`, we utilize the scheduling sampling strategy (Liu et al., 2021) for fine-tuning further.

For the other three domains, we do not use Step 2 because we do not have any other in-domain data.

2.6 Post Editing

Post editing is a way to correct significant errors in the model translation. In all directions, we insist on using common rules to correct significant errors in the model translation. The errors include:

- Misalignment of symbols and emoji between source and target languages.

- The unnecessary space between Url, HTML, and the text in parenthesis.

In the final submission, this process corrects approximately 2% of all tokens in the test set (most of them are symbols such as extra Spaces between characters).

3 Experiment

3.1 Experiment Settings

The implementation of our models was based on `fairseq` (Ott et al., 2019), and the total data we used were shown in Table 3. In the ZH→EN direction, All models were trained on 8 RTX 2080Ti GPUs, and all other direction models were trained on 4 RTX 3090 GPUs. We used Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.997$ during training. Except for the base model, all of our models adopted the pre-norm structure (Wang et al., 2019). Following the idea of the work (Wang et al., 2019), we adopted the deep and wide model to increase the model capacity (Zhou et al., 2021; Wang et al., 2019). Under the GPU memory constraint, we accumulated the gradient four times and set the batch size to 2048 tokens.

For the deep model, we trained the model for 15 epochs at most. We set max learning rate = 0.002 and warmup step = 8000 for all deep models. All dropout probabilities were 0.1. Meanwhile, we also used FP16 to accelerate the training process. All experiments were evaluated on `newstest2021` using `SacreBLEU` (Post, 2018) in the EN→HR and the EN↔LIV directions. In the ZH→EN direction, we used `multi-bleu.perl`[3]. At last, we introduced a patience factor during decoding, which provided a more flexible decoding depth (Kasai et al., 2022). However, this method led to a significantly slower decoding speed. So we only applied this method to generate the final output.

[3] https://github.com/moses-smt/mosesdecoder

Model	depth	P&R	Head	Hidden Size	Filter Size	Batch size	update freq
BASE	6	✗	8	512	2048	4096	2
RPR	24	✓	8	512	4096	2048	4
DLCL	25	✓	8	512	4096	2048	4
ODE	6	✓	16	1024	4096	1024	16
ODE	12	✓	16	1024	4096	1024	16
UMST	24	✓	8	512	4096	2048	4

Table 2: Explicit information of model structure, P&R indicates whether to use the pre-Norm and relative position representation(RPR)

	Bilingual	Monolingual	
		EN	Other
ZH→EN	12.10M	8M	11M
EN→HR	46M	5M	20M
EN↔LIV	600	0.15M	0.04M

Table 3: The sentences we use in each direction after filtering(The M stands for million).

3.2 ZH→EN

For the ZH↔EN tasks, we only submit in the ZH→EN direction. We filter out the part of data from ParaCrawl, News Commentary V16, WikiMatrix, UN Parallel Corpus V1.0, and the CCMT Corpus as the training set. We use the filtering rules and XenC mentioned above for data filtering. We end up with 12 million raw bilingual data as the training set.

Regarding the multi-domain adaptation, we try an unsupervised data clustering method that uses the pre-trained model's hidden state to do domain classification in the training set. We also use TF-IDF to select keywords from the test set to represent each domain and then use these keywords to select in-domain sentences from constrained data. Unfortunately, the aforementioned methods show poor performance except in the news domain. We find no significant domain difference between the constrained bilingual data and constrained monolingual data we used.

Based on the training set, we train several deep models mentioned above. We use `newstest2020` as the valid set and `newstest2021` as the test set to modify the hyper-parameters and find the optimal ensemble combination. In addition, we also realize a multi-domain translation model which introduces layer-wise Domain Mixing into the vanilla Transformer. However, the model performs poorly on the inseparable domain data, so it is not included in our model ensemble.

For the first round of back-translation, we filter multiple groups' English monolingual data from the News crawl, News discussions, Europarl v10, News Commentary, Common Crawl, and Leipzig Corpora. The amount of data is about 4 million to 8 million sentences. We use the best ensemble model to translate monolingual data with Beam Search, Top-K, and Top-P decoding. By directly concatenating the raw training and pseudo data, we fine-tune the existing model and achieve +0.85 BLEU improvement after the first back-translation iteration, then achieve +0.59 BLEU improvement after the second back-translation iteration.

For knowledge distillation, we filter 3 million monolingual data from News crawl, News Commentary, Common Crawl, Extended Common Crawl, and Leipzig Corpora. We use the best ensemble model to translate the monolingual and obtain the pseudo data, and then fine-tune each sub-model in the ensemble model. We obtain the improvement of 0.16 BLEU points. We select the best four models to construct the ensemble model every time during back-translation and knowledge distillation in both directions.

For fine-tuning, we first do fine-tuning on the news domain to search the optimal hypo-parameters. We use `newstest2019`, `newstest2020` in both ZH→EN and EN→ZH directions as the training set, and obtain the optimal learning rate of 0.001 on `newstest2021`. We achieve +0.93 BLEU improvement in the ZH→EN direction. Then we add `newstest2021` to the training set for fine-tuning. In order to improve the performance of the model in a single domain, we divide the `test2022` into five domains according to the domain labels: news, social, conversational, e-commerce, and biomedical. Finally, we do domain adaptation separately in four domains except biomedical by fine-tuning the model with scheduling sampling.

At last, we use four single-domain models to generate translation in every single domain and use post-processing methods to correct the error in the translation, which brings us +0.81 BLEU improvement in the ZH→EN direction. Our experimental results are shown in Table 4.

3.3 EN→HR

For the EN→HR tasks, we choose ParaCrawl v9, Tilde MODEL corpus, WikiMatrix, and OPUS total of four parallel data corpora of about 90M. We choose all of the News Crawl and Leipzig Corpora for the Croatian monolingual data of about 20M. In order to strengthen the generalization of the model in the social, conversational, and e-commerce domains, we choose the Web and Wikipedia parts from Leipzig Corpora about 10M for the English monolingual data to distill our models.

In addition to the common data filtering process, we calculate the Levenshtein ratio of two adjacent sentences from sorted sentences to remove duplication sentences whose Levenshtein ratio are not less than 0.85. After the data filtering, about 46M sentence pairs are left to build our system. Additionally, we use a shared vocabulary and set the merge operations of BPE to 32K.

Since the domain of the official development set focuses on e-commerce and reviews, we make a general domain test set by ourselves to evaluate the model generalization ability better. To use the Croatian monolingual data, we implement tagged back-translation, which brings us +0.35 BLEU improvement on the official development set and +0.5 BLEU improvement on our test set. We also implement knowledge distillation to use English monolingual data, which brings us +0.3 BLEU improvement on the official development set and +0.14 BLEU improvement on our test set.

We use XenC to select 5M sentence pairs similar to the official development set from the original training set and then fine-tune each model for several epochs. However, we find that not only fine-tuning significantly reduces the model's generalization, but also has a slightly better performance on the official development set and significantly worse performance on our test set. Finally, we put all models together to search for the best ensemble greedily. This method brings us +0.51 BLEU improvement on the official development set and +0.25 BLEU improvement on our test set.

During post-processing, we use rules to adjust the order of punctuation, case inconsistencies and remove some extra spaces, which brings us +0.43 BLEU improvement on the official development set.

3.4 EN↔LIV

For the EN↔LIV tasks, we create a many-to-many multilingual submission for WMT2022. The multilingual submission includes seven language directions, which are CS→EN, ET→EN, LV→EN, EN↔LIV, ET→LIV, and LV→LIV. For CS→EN , we only use ParaCrawl v9 dataset and obtain 50M parallel data after cleaning. After the data filtering, we sample the top 10M data according to a language model trained with CS→EN data. For ET→EN, LV→EN, EN↔LIV, ET→LIV, and LV→LIV directions, we only use OPUS liv4ever v1 dataset, separately obtaining 956, 997, 540, 11420, 10786 parallel data after cleaning. We use the valid set and test set in OPUS liv4ever v1 data set as our valid set and test set. It is worth noting that we delete the same sentences in the test set and the train set.

We use a combination of multiple language directions to train the baseline model, including many-to-many and many-to-one, and find that models trained by all language directions data and many-to-many is 1 BLEU point higher on average than the model trained by several language directions data or many-to-one in the test set. We find that data in different language directions can provide semantic help to EN↔LIV model because CS, LV, ET and LIV are similar languages. So, we select all language directions data and many-to-many to train our model.

We also use pre-trained model for language modeling. Since the constrained track, we choose the AB-Net (Guo et al., 2020) model whose encoder and decoder are initialized with mBERT. However, the performance of AB-Net model was lower than that of the baseline model, so it is not included in our final submission results. The poor performance may be due that: first, the pre-trained model doesn't contain LIV, and second, the parallel data of EN↔LIV is too scarce. This leads to a big challenge to transform the knowledge of the pre-trained model into the EN↔LIV translation model.

Due to the lack of EN↔LIV parallel data, the model cannot capture the alignment information at the word level. Therefore, we make a parallel

System	ZH→EN	EN→HR	EN→LIV	LIV→EN
Baseline	24.27	31.28	4.1	6.57
Deep model	27.2	32.68	5.66	8.79
+ Dict	–	–	8.16	13.79
+ Iteratively BT	28.64	33.03	–	–
+ Iteratively KD	28.8	33.33	8.96	15.99
+ Fine-tuning	29.73	–	10.26	16.89
+ Ensemble	-	33.84	10.48	16.95
+ Post edit	30.54	34.27	–	–

Table 4: BLEU evaluation results on the WMT 2021 ZH→EN, EN↔LIV test sets and WMT 2021 EN→HR development sets.

dictionary of EN↔LIV. First, we use fast_align [4] tool to align the words on the EN↔LIV dataset, and then manually check and modify it. Finally, we obtain a parallel dictionary of EN↔LIV with a dictionary size of 3127. We mix the parallel dictionary and parallel data of EN↔LIV to obtain new parallel data. Then, we train the model by new parallel data and bring us +5 BLEU improvement in the LIV→EN direction and +2.5 BLEU improvement in the EN→LIV direction.

We also use iterative back-translation and iterative knowledge distillation to enhance the model. Since the many-to-many method, the back-translation implemented in the LIV→EN direction is the same as the knowledge distillation in the EN→LIV direction. During the back-translation on the EN→LIV direction, we use 40000 LIV monolingual data from OPUS liv4ever v1 data set. And then during the knowledge distillation on the EN→LIV direction, we use the test set in OPUS liv4ever v1 as in-domain data, and we use the XenC tool to sample 150000 EN monolingual data from Europarl v10 based on in-domain data. We generate pseudo data by using both post-ensemble and ensemble methods. We obtain the improvement of 2.2 BLEU points and 0.8 BLEU points in the back-translation (knowledge distillation) in LIV→EN and EN→LIV. After KD, we use the OPUS liv4ever v1 valid set to fine-tune our models for five epochs with the 0.0003 learning rate and obtain +0.9 and +1.3 BLEU improvement in the LIV→EN and EN→LIV directions.

3.5 Submission Results

The results of our best submissions in four directions this year are shown in Table 5. In the EN→HR direction, our system performed well trained on large amounts of bilingual data. In the EN↔LIV direction, our multilingual model performance is better than the model initialized by the pre-trained model(e.g., mBERT), indicating that the multilingual model has potential in the low resource language. In the ZH→EN direction, KD is not performing well enough in `newstest2021` as usual. On the one hand, this may be related to our data filtering method and the domain changes on the test set; on the other hand, it may be related to our stronger deep model.

Direction	Submission
ZH→EN	26.2
EN→HR	18.1
EN→LIV	12.3
LIV→EN	13.0

Table 5: Our final submission results in four directions.

4 Conclusion

This paper introduces our submissions on WMT22 in four directions. We train our system with constrained data in all directions. The system is constituted by the ensemble model based on multiple deep models.

For training data, we use a softer data filtering method to obtain more data and make the model more robust in the general domain. Based on this data, model performance is better than our last year's systems. We use iterative back-translation and knowledge distillation methods which have been proven to be very effective in the past. In addition, fine-tuning using both normal training and scheduling sampling also achieves good results.

In the ZH→EN direction, we mainly build the news domain translation model. Also, we try the multi-domain data clustering method and

[4] https://github.com/clab/fast_align

multi-domain adaptation method to build the multi-domain model. However, because the sentences in constrained data have no noticeable domain difference, the performance of the above method is not satisfactory. In the EN↔LIV direction, we try the multilingual model and initialization method, which initialize the translation model with the pre-trained model. We find that the multilingual model show more considerable potential than the model initialized with mBERT, even under minimal data.

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7747–7763. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 1: Research Papers*, pages 53–63. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 644–648. The Association for Computational Linguistics.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *CoRR*, abs/1702.01802.

Junliang Guo, Zhirui Zhang, Linli Xu, Hao-Ran Wei, Boxing Chen, and Enhong Chen. 2020. Incorporating BERT into parallel sequence decoding with adapters. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Cong Duy Vu Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, NMT@ACL 2018, Melbourne, Australia, July 20, 2018*, pages 18–24. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Haoming Jiang, Chen Liang, Chong Wang, and Tuo Zhao. 2020. Multi-domain neural machine translation with word-level adaptive layer-wise domain mixing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1823–1834. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Trans. Assoc. Comput. Linguistics*, 5:339–351.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Dragomir R. Radev, Yejin Choi, and Noah A. Smith. 2022. Beam decoding with controlled patience. *CoRR*, abs/2204.05424.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1317–1327. The Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Bei Li, Quan Du, Tao Zhou, Shuhan Zhou, Xin Zeng, Tong Xiao, and Jingbo Zhu. 2021. ODE transformer: An ordinary differential equation-inspired model for neural machine translation. *CoRR*, abs/2104.02308.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The niutrans machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 257–266. Association for Computational Linguistics.

Bei Li, Tong Zheng, Yi Jing, Chengbo Jiao, Tong Xiao, and Jingbo Zhu. 2022. Learning multiscale transformer models for sequence generation. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 13225–13241. PMLR.

Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. 2021. Scheduled sampling based on decoding steps for neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3285–3296. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign@IWSLT 2015, Da Nang, Vietnam, December 3-4, 2015*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 48–53. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 186–191. Association for Computational Linguistics.

Anthony Rousseau. 2013. Xenc: An open-source tool for data selection in natural language processing. *Prague Bull. Math. Linguistics*, 100:73–82.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 464–468. Association for Computational Linguistics.

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. Facebook ai's WMT21 news translation task submission. In *Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP 2021, Online Event, November 10-11, 2021*, pages 205–215. Association for Computational Linguistics.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1810–1822. Association for Computational Linguistics.

Tong Xiao, Jingbo Zhu, Hao Zhang, and Qiang Li. 2012. Niutrans: An open source toolkit for phrase-based and syntax-based machine translation. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the System Demonstrations, July 10, 2012, Jeju Island, Korea*, pages 19–24. The Association for Computer Linguistics.

Xianfeng Zeng, Yijin Liu, Ernan Li, Qiu Ran, Fandong Meng, Peng Li, Jinan Xu, and Jie Zhou. 2021. Wechat neural machine translation systems for WMT21. In *Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP 2021, Online Event, November 10-11, 2021*, pages 243–254. Association for Computational Linguistics.

Yuhao Zhang, Ziyang Wang, Runzhe Cao, Binghao Wei, Weiqiao Shan, Shuhan Zhou, Abudurexiti Reheman, Tao Zhou, Xin Zeng, Laohu Wang, Yongyu Mu, Jingnan Zhang, Xiaoqian Liu, Xuanjun Zhou, Yinqiao Li, Bei Li, Tong Xiao, and Jingbo Zhu. 2020. The niutrans machine translation systems for WMT20. In *Proceedings of the Fifth Conference on Machine Translation, WMT@EMNLP 2020, Online, November 19-20, 2020*, pages 338–345. Association for Computational Linguistics.

Shuhan Zhou, Tao Zhou, Binghao Wei, Yingfeng Luo, Yongyu Mu, Zefan Zhou, Chenglong Wang, Xuanjun Zhou, Chuanhao Lv, Yi Jing, Laohu Wang, Jingnan Zhang, Canan Huang, Zhongxiang Yan, Chi Hu, Bei Li, Tong Xiao, and Jingbo Zhu. 2021. The niutrans machine translation systems for WMT21. In *Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP 2021, Online Event, November 10-11, 2021*, pages 265–272. Association for Computational Linguistics.

Teaching Unseen Low-resource Languages to Large Translation Models

Maali Tars, Taido Purason, Andre Tättar
TartuNLP, University of Tartu
maali.tars@ut.ee, taido.purason@ut.ee, andre.tattar@ut.ee

Abstract

In recent years, large multilingual pre-trained neural machine translation model research has grown and it is common for these models to be publicly available for usage and fine-tuning. Low-resource languages benefit from the pre-trained models, because of knowledge transfer from high- to medium-resource languages. The recently available M2M-100 model is our starting point for cross-lingual transfer learning to Finno-Ugric languages, like Livonian. We participate in the WMT22 General Machine Translation task, where we focus on the English-Livonian language pair. We leverage data from other Finno-Ugric languages and through that, we achieve high scores for English-Livonian translation directions. Overall, instead of training a model from scratch, we use transfer learning and back-translation as the main methods and fine-tune a publicly available pre-trained model. This in turn reduces the cost and duration of training high-quality multilingual neural machine translation models.

1 Introduction

We participate in the WMT 2022 General Machine Translation shared task where we submit a system for English-Livonian and Livonian-English translation directions. Our system is trained in the unconstrained setting utilizing data from other languages that are all in a way related to Livonian.

Recently, the development of large multilingual models has been increasing (Johnson et al., 2017; Gu et al., 2018; Fan et al., 2021; NLLB Team et al., 2022) and thus there are multiple pre-trained multilingual models available for further fine-tuning to a specific task. Fine-tuning these models on in-domain data saves time and computational costs by not having to train a multilingual model from scratch. We utilize the M2M-100 multilingual pre-trained neural machine translation (NMT) model (Fan et al., 2021) and do cross-lingual transfer learning to low-resource language pairs from the Finno-Ugric language family, including the Livonian language. We further improve our model with two back-translation iterations and a final fine-tuning on languages that have available original parallel data paired with Livonian.

The languages we use to support the English (en)-Livonian (liv) directions are from the Finno-Ugric language family or geographically close to that family of languages: Finnish (fi), Estonian (et), Latvian (lv), Norwegian (no), Võro (vro), North Sami (sme), South Sami (sma), Inari Sami (smn), Skolt Sami (sms), Lule Sami (smj).

The structure of the article consists of giving insight into the related work in the field of low-resource NMT and from the Finno-Ugric language family perspective in Section 2, the description of data in Section 3, the overview of our system architecture and training methods in Section 4, description of experiments in Section 5 and the results in Section 6.

2 Related work

2.1 Low-resource setting

There have been a lot of efforts in trying to achieve high-quality translation for low-resource languages in order for them to catch up with high- and medium-resource languages. The main benefits seem to come from performing transfer learning to low-resource languages with previous knowledge acquired from a high-resource language (Gu et al., 2018).

Another aspect is data augmentation. Commonly, low-resource languages have a lot more monolingual data available than parallel data, which enables producing synthetic parallel samples that have been shown to improve the accuracy of translation (Sennrich and Zhang, 2019).

For the Finno-Ugric languages, in Rikters et al. (2018), they note that in efforts of achieving better translation quality for Estonian, training a multi-

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 375 - 380
December 7-8, 2022 ©2022 Association for Computational Linguistics

lingual model gets the best result. It usually helps even more if the high- or medium-resourced languages in the mix during training are closely related to the low-resource languages as shown in Tars et al. (2021). In Kocmi and Bojar (2018), the authors proved transfer learning to be very beneficial for languages with low amounts of parallel resources. However, in some cases, they saw more improvements when the high-resource language was not related to the low-resource language.

2.2 M2M-100

M2M-100 is a massively multilingual pre-trained machine translation model featuring many-to-many translations between 100 languages (Fan et al., 2021). It was trained on 7.5 billion parallel sentence pairs which, unlike datasets for many previous approaches, were chosen to make the dataset non-English-Centric. Fan et al. (2021) were able to compose the non-English-Centric training dataset through the use of bitext mining and back-translation. The improvement of M2M-100 over previous models is especially visible in non-English directions and low-resource languages. The vast amount of training data, many supported languages, and promising results reported by Fan et al. (2021) give us reason to believe that M2M-100 would be also a good starting point for training a Finno-Ugric system.

3 Data

3.1 Additional languages

We did not limit ourselves to only English-Livonian training data, because the amount of parallel data for that language pair seemed too scarce to train a quality machine translation system. Instead, we decided to leverage our previous research into Finno-Ugric languages (Tars et al., 2021) and include the language pairs that are closely related to Livonian grammatically as well as geographically.

We added four languages that were high- or medium-resource: Estonian, Finnish, Latvian, Norwegian. The aim of including these languages was for them to aid the low-resource Finno-Ugric languages in the training process. The low-resource languages that we included were: Võro, North Sami, South Sami, Inari Sami, Skolt Sami, Lule Sami.

As Livonian has historically been spoken mainly in the areas that are nowadays Latvia, its language has shaped Livonian noticeably, even though Latvian itself is part of the separate Baltic branch of languages. Multiple low-resource languages that we also included are Sami languages, which are mainly spoken in the areas of Norway, Sweden and Finland. Most of the parallel data available for Sami languages is paired with either Finnish or Norwegian. Norwegian is not part of the Finno-Ugric language family, but as was the case for Latvian, it is spoken in the same area as some of the Sami languages and has influenced them over time, for example sharing some orthographic symbols in the vocabulary.

3.2 Pre-processing and filtering

The data not provided by the shared task was collected from various openly available sources, such as META-SHARE[1] and translation memory compiled by the Arctic University of Norway[2]. Further details about the data sources are described in Tars et al. (2021). We compiled all of the filtered parallel data and the monolingual data and publish it on our HuggingFace page [3].

Following the collection phase, we applied multiple pre-processing and filtering heuristics to the parallel data, as well as deduplicated the whole dataset. We normalized punctuation and detokenized the data with the help of Moses scripts, however, we modified the normalization script for it to be more applicable to Finno-Ugric languages[4]. Detokenization language code defaulted to English if the script did not recognize the language code of a low-resource language. Filtering and whitespace normalization was done with the OpusFilter tool (Aulamo et al., 2020). We provide a list of filters used:

- maximum segment length: 1000 characters or 400 words
- maximum word length: 50 characters
- source and target segment length difference: max 3 times
- ratio of numeric characters in segment: 0.5 or less

[1] https://doi.org/10.15155/1-00-0000-0000-0000-001A0L

[2] https://giellalt.uit.no/tm/TranslationMemory.html

[3] https://huggingface.co/datasets/tartuNLP/finno-ugric-train

[4] https://github.com/Project-MTee/model_training/blob/main/normalization.py

lang-pair	et-vro	fi-sme	fi-sma	fi-smn	fi-sms	no-sma	no-sme	no-smj	sme-sma	sme-smj	sme-smn	en-liv	et-liv	lv-liv
filtered	29 775	62 837	2766	9459	2708	15 702	195 970	11 627	19 963	14 985	894	280	12 887	10 763

Table 1: Parallel data numbers after filtering (in sentence pairs).

language	vro	sma	sme	sms	smn	smj	liv
nr of segments	162 807	55 088	33 964	76 685	122 916	128 180	40 329

Table 2: Monolingual data numbers.

- ratio of alphabetic characters in latin alphabet: 1
- ratio of alphabetic characters in segment: 0.75 or more
- ratio of similar numerals between segments, with zeros removed: 0.5 or more

Some of the values are default from OpusFilter, but others had to be tuned to filter out the noisy training samples that were left undetected with the default parameters. The data numbers of all the parallel data for all of the translation directions left after filtering can be seen in Table 1. Additionally, we sampled 20 000 segments from corpora available in OPUS (Tiedemann, 2012) for each language pair between high- to medium-resource languages (et, en, lv, no, fi).

3.3 Monolingual data

We also gathered monolingual data for all the languages involved. The monolingual data for high- and medium-resourced languages was acquired from previously available WMT sources. For the low-resource languages, the data was scraped from various files from the web, that were collected either by the Arctic University of Norway or ourselves.

We sampled 500 000 random segments for all of the high- to medium-resource languages from publicly available data (et, en, lv, no, fi). The amounts of monolingual data for low-resource languages can be seen in Table 2. No filtering was done to the monolingual data, but the data segments all went through the same detokenization and normalization scripts that were applied to parallel data. After the back-translation iterations explained in Section 4.2, the synthetic parallel samples were also not filtered.

For the English-Livonian directions, the only parallel and monolingual data used was the data provided by the WMT.

3.4 Evaluation benchmarks

In order to evaluate the multiple translation directions we had other than English-Livonian, we created new test sets[5] for them, because there are no publicly available benchmarks for translation directions like Finnish-Inari Sami, for example. The test sets are composed of held-out data from the parallel datasets. For all the language pairs containing at least one of the low-resource languages, we extracted 500 sentences for evaluation and 200 sentences for validation.

4 System overview

4.1 M2M-100 settings

Our final system builds on the large pre-trained multilingual neural machine translation model M2M-100. Livonian along with other low-resource Finno-Ugric languages were not part of the training process of M2M-100. We use the HuggingFace implementation of M2M-100[6]. Fine-tuning this model for previously unseen languages requires introducing new symbols to the vocabulary and increasing the embedding matrix. We created scripts[7] that allow expanding the embedding matrix of a pre-trained model and thus make it possible to do cross-lingual transfer learning.

4.2 Stages of training

This section describes the training of our final system. The first stage of transfer learning used all of the original Finno-Ugric parallel data that we had. We decided to go with the M2M-100's 1.2 billion parameter model (1.2B) as our starting point because our previous experiments showed that it improves more than the smaller, 418 million parameter model (418M) on the data that we have (Tars et al., 2022).

[5] https://huggingface.co/datasets/tartuNLP/finno-ugric-benchmark

[6] https://huggingface.co/docs/transformers/model_doc/m2m_100

[7] https://github.com/TartuNLP/m2m-100-finetune

After that, we performed the first iteration of back-translation with all of the monolingual data. Combining the original parallel data and the synthetic data, we fine-tuned the M2M-100 1.2B model again and performed the second iteration of back-translation with the newly fine-tuned model. The monolingual data stayed the same.

For the next step we went back to do transfer learning from the beginning on the 1.2B model, but this time the data we used consisted of the original parallel data and the data produced in the second iteration of back-translation, leaving the data from the first iteration out. Finally, we fine-tuned the model on original parallel data for language pairs between `en-liv-et-lv`.

5 Experiments

5.1 Experimental settings

All our systems, including the final system, were trained on one Tesla A100 GPU with 40GB vRAM. Our experiments were done on two versions of the M2M-100 model: 418M model and 1.2B model. The learning rate was initialized with the default value from HuggingFace code. Batch size was 12 with gradient accumulation steps set to 8.

5.2 Different experiments

The size of the model was one aspect of experimentation that we looked into. As smaller models are easier and quicker to fine-tune and deploy, comparing the 418M and 1.2B models seemed necessary. 1.2B model has more parameters, but the intuition was that maybe the 418M model is also big enough for this specific dataset, because it is relatively small.

The main approach to enhance `en-liv` results was leveraging information from other Finno-Ugric languages. We trained models on all the Finno-Ugric language data described, as well as dividing the languages into even smaller groupings, as described in Tars et al. (2022). Subsequently, we performed additional experiments to see whether the added languages really help the Livonian language.

We repeated the stages of training described in Section 4.2 but with different-sized models and with a smaller dataset, consisting only of languages paired with Livonian.

	COMET-A ↓	ChrF-all
`en-liv`	-36.8	39.2
`liv-en`	-5.8	53.5

Table 3: Automatic metric results of our primary system on WMT22 test set.

6 Results

6.1 Automatic metrics

According to the automatic metric results, our system performs the best in the Livonian-English translation direction and achieves second place in the English-Livonian direction. The metrics that were used were COMET and ChrF. The results of the automatic evaluation can be seen in Table 3.

During the development period, we measured most of our additional experiments on BLEU. The results of those experiments compared to the earlier results for English-Livonian translation directions can be seen in Table 4. For further understanding of where the gain in performance happened, we describe the results of intermediate models that were trained before arriving at the final system.

Firstly, we can observe that `en-liv` results are about half of the `liv-en` results and that the BLEU score improvements come from different techniques for either of the translation directions. For `en-liv`, the main source of improvement is the last stage of fine-tuning the model on the original parallel `en-et-lv-liv` data. For `liv-en` however, the biggest gain happens with back-translation. This could be explained by the amount of monolingual data, as Livonian had only about 40 000 segments but for English, we sampled 500 000 segments.

Another aspect we can point out is the relatively small difference between the smaller (418M) and the larger (1.2B) model results. The 1.2B model is better at every stage as expected, but considering how much more computational cost and deployment resources the larger model requires, the trade-off in quality might be tolerable.

Lastly, compared to the previous best results reported by Rikters et al. (2022), our models surpass those results by about 4 BLEU for `en-liv` and 12 BLEU for `liv-en`.

6.2 Results for other language pairs

Additionally, we report results on our held-out test set described in Section 3.4 for low-resource language pairs that were a part of our final system development. The results can be seen in Table

	`en-liv`	`liv-en`
1.2B (baseline)	10.15	18.92
+ bt1	11.24	28.67
+ bt2	12.16	29.37
+ tuned on `liv`	**15.19**	**31.06**
+ bt1 only-`liv`	10.66	27.88
+ bt2 only-`liv`	11.21	29.85
+ tuned on `liv`	11.56	30.33
Rikters et al., 2022	*11.03*	*19.01*

	`en-liv`	`liv-en`
418M (baseline)	10.29	15.78
+ bt1	11.25	27.52
+ bt2	10.62	27.38
+ tuned on `liv`	**12.83**	27.23
+ bt1 only-`liv`	11.39	27.74
+ bt2 only-`liv`	11.63	28.81
+ tuned on `liv`	11.53	**29.27**
	11.03	*19.01*

Table 4: Experiment results on BLEU. "1.2B" and "418M" refer to models trained with all original parallel data. "bt1" is trained on parallel + first back-translation iteration data, "bt2" on parallel + second back-translation iteration data. "only-`liv`" - only data between `et-en-lv-liv` languages was used for training. "tuned on `liv`" refers to the "bt2" model that was tuned on `et-en-lv-liv` original parallel data. Last row represents previously best results for `en-liv-en` by Rikters et al. (2022).

5 and Table 6. The language pairs were evaluated on the final system and although the final system was chosen on `en-liv-en` validation data, we see good overall results for other low-resource language pairs as well. However, the results in Table 5 are significantly lower than the results reported in Tars et al. (2022) on the same test data. This is probably caused by the fact that as the last training stage, the final system was fine-tuned only on `et-en-lv-liv` original parallel data.

For `et-liv-et` and `lv-liv-lv` directions, however, we report new state-of-the-art results on the test data that was also used in Rikters et al. (2022).

7 Conclusion

Large pre-trained multilingual neural machine translation models prove to be beneficial to low-resource Finno-Ugric languages, such as Livonian. We placed in the top 2 for the English-Livonian language pair in the WMT22 General Machine Translation shared task. Training in an unconstrained setting gets reasonable and good-quality results, especially when using languages close to Livonian to help achieve a better translation quality. In the future, we plan to test additional and more recent pre-trained multilingual models as a starting point for cross-lingual transfer learning and add more low-resource Finno-Ugric languages into the dataset.

Limitations

The 1.2B M2M-100 model has a lot of parameters which makes deploying this model very costly and difficult because it needs a lot of memory and is computationally unfeasible. In turn, it also makes the training somewhat slower in terms of loading the model parameters and updating them. We are working on trying to reduce the vocabulary and number of parameters, by removing parts of the vocabulary not necessary for Finno-Ugric languages. Another thing we left out of the process was filtering monolingual and synthetic data, which might be a useful addition to the pre-processing pipeline.

References

Mikko Aulamo, Sami Virpioja, and Jörg Tiedemann. 2020. OpusFilter: A configurable parallel corpus filtering toolbox. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 150–156, Online. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O.K. Li. 2018. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 344–354, New Orleans, Louisiana. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

	WMT22 sys	*previous*
`fi-sma-fi`	12.31	***29.04***
`fi-sme-fi`	36.88	***46.56***
`fi-smn-fi`	53.73	***64.37***
`fi-sms-fi`	36.14	***47.61***
`no-sma-no`	40.59	***50.16***
`no-sme-no`	33.89	***40.61***
`no-smj-no`	32.31	***46.22***
`sme-sma-sme`	26.16	***40.37***
`sme-smj-sme`	22.32	***40.24***
`sme-smn-sme`	31.73	***33.88***
`et-vro-et`	34.76	***37.08***

Table 5: BLEU scores for low-resource language pairs included in the final system. *previous* signifies the previous best results for these language pairs reported in Tars et al. (2022).

	WMT22 sys	*previous*
`et-liv`	**18.31**	*16.49*
`liv-et`	**24.00**	*23.05*
`lv-liv`	**19.16**	*17.65*
`liv-lv`	**26.33**	*25.24*

Table 6: BLEU scores for low-resource language pairs included in the final system. *previous* signifies the previous best results for these language pairs reported in Rikters et al. (2022).

Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252, Brussels, Belgium. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia-Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Matīss Rikters, Mārcis Pinnis, and Rihards Krišlauks. 2018. Training and adapting multilingual NMT for less-resourced and morphologically rich languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Matīss Rikters, Marili Tomingas, Tuuli Tuisk, Valts Ernštreits, and Mark Fishel. 2022. Machine translation for Livonian: Catering to 20 speakers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 508–514, Dublin, Ireland. Association for Computational Linguistics.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

Maali Tars, Andre Tättar, and Mark Fišel. 2021. Extremely low-resource machine translation for closely related languages. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 41–52, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Maali Tars, Andre Tättar, and Mark Fišel. 2022. Cross-lingual transfer from large multilingual translation models to unseen under-resourced languages. *Baltic Journal of Modern Computing*, 10.3:435–446.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Can Domains Be Transferred Across Languages in Multi-Domain Multilingual Neural Machine Translation?

Thuy-Trang Vu[◇*] and **Shahram Khadivi**[†]
Xuanli He[◇] and **Dinh Phung**[◇] and **Gholamreza Haffari**[◇]
[◇]Department of Data Science and AI, Monash University, Australia
[†] eBay Inc.
`{trang.vu1,xuanli.he1,first.last}@monash.edu`
`skhadivi@ebay.com`

Abstract

Previous works mostly focus on either multilingual or multi-domain aspects of neural machine translation (NMT). This paper investigates whether the domain information can be transferred across languages on the composition of multi-domain and multilingual NMT, particularly for the incomplete data condition where in-domain bitext is missing for some language pairs. Our results in the curated leave-one-domain-out experiments show that multi-domain multilingual (MDML) NMT can boost zero-shot translation performance up to +10 gains on BLEU, as well as aid the generalisation of multi-domain NMT to the missing domain. We also explore strategies for effective integration of multilingual and multi-domain NMT, including language and domain tag combination and auxiliary task training. We find that learning domain-aware representations and adding target-language tags to the encoder leads to effective MDML-NMT.

1 Introduction

Multilingual NMT (MNMT), which enables a single model to support translation across multiple directions, has attracted a lot of interest both in the research community and industry. The gap between MNMT and bilingual counterparts has been reduced significantly, and even for some settings, it has been shown to surpass bilingual NMT (Tran et al., 2021). MNMT enables knowledge sharing among languages, and reduces model training, deployment, and maintenance costs. On the other hand, multi-domain NMT aims to build robust NMT models, providing high-quality translation on diverse domains. While multilingual and multi-domain NMT are highly appealing in practice, they are often studied separately.

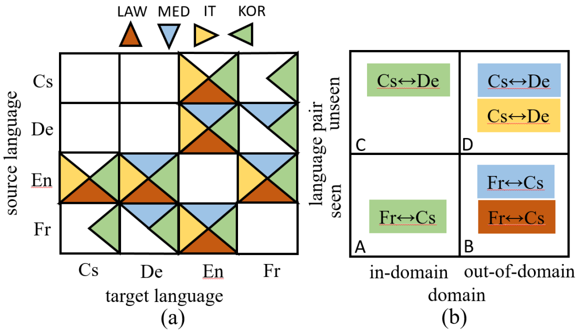


Figure 1: An example of the multi-domain multilingual incomplete data condition (best seen in colours). (a) The colour indicates the availability of bitext in the corresponding domain for each language. (b) Domain and language-pair matrix for the data condition in (a).

To accommodate the domain aspect, previous MNMT works focus on learning a domain-specific MNMT by finetuning a general NMT model on the domain of interest (Tran et al., 2021; Bérard et al., 2020). Recently, Cooper Stickland et al. (2021) propose to unify multilingual and multi-domain NMT into a holistic system by stacking language-specific and domain-specific adapters with a two-phase training process. Thanks to the plug-and-play ability of adapters, their system can handle translation across multiple languages and support multiple domains. However, as each domain adapter is learned independently, their adapter-based model lacks the ability of effective knowledge sharing among domains.

In this paper, we take a step further toward unifying multilingual and multi-domain NMT into a single setting and model, *i.e.,* multi-domain multilingual NMT (MDML-NMT), and enable effective knowledge sharing across both domains and languages. Unlike the *complete* data assumption in the multi-domain single language-pair setting where training data is available in all domains, we assume the existence of bitext in all domains for only a subset of language-pairs, as illustrated in Figure 1(a). In fact, it is highly improbable to obtain in-domain bitext for all domains and all language pairs in

[*]Work done while doing internship at eBay Inc.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 381 - 396
December 7-8, 2022 ©2022 Association for Computational Linguistics

many real-life settings. Depending on the availability of parallel data, we categorise a translation task from a source to a target language into four categories based on the following dimensions:

- *in-domain*/*out-of-domain*, wrt to the domain of interest, and
- *seen*/*unseen*, wrt to the translation direction during training.

Please note the domain and language-pair matrix in Figure 1(b). In this figure, parallel data available in the training set specifies the group A, the *in-domain seen* tasks. Given this training dataset, most MNMT research focuses on cross-lingual transfer to *in-domain unseen* translation tasks (A→C), while the studies on multi-domain NMT and domain adaptation seek to generalise to *out-of-domain seen* translation tasks (A→B). Integrating domain and language aspects in the incomplete data condition gives rise to an interesting and more challenging setting that transfers to *out-of-domain unseen* translation tasks (A→D). We hypothesise that the out-of-domain "seen and unseen" translation tasks (A→B+D) can benefit from the in-domain translation tasks if there exists the domain transfer across languages in MDML-NMT.

Specifically, we ask the following research questions: (1) Do out-of-domain translation tasks benefit from the out-of-domain and in-domain bitext in other seen translation pairs? and (2) What is effective method to handle the composition of domains and languages? Furthermore, beyond the cross-lingual transfer (A→C) and the out-of-domain generalisation (A→B), we also consider the challenging setting where the translation direction of interest may not have any bitext in any domain, i.e. the zero-shot setting (A→D).

In general, we can vary the degree of domain transfer based on the number of domains in which parallel data for a translation task is available. Combining with the number of language pairs of interest, there are large numbers of incomplete data conditions, even for our toy examples in Figure 1. In this study, we assume the highest degree of domain transfer and carefully design controlled experiments where one domain is left out for some language pairs (Table 1). We then examine the potential of MDML-NMT on this incomplete data condition. We also explore training strategies for effective integration of multi-domain and multilingual NMT, mainly on (i) how to combine the language and domain tags, and (ii) using auxiliary task training to learn effective representations. Our contributions are as follows:

	LAW	IT	KORAN	MED	SUB
En-Fr	✔	✔	✔	✔	✔
En-De	✔	✔	✔	✔	✔
De-Fr	✔	✔	✔	✔	✔
En-Cs	✗	✔	✔	✔	✔
En-Pl	✗	✔	✔	✔	✔

Table 1: Illustration of leave-one-out LAW experiment setting. ✗, ✔describes whether there is bitext in the corresponding domain for the given language pairs.

- We investigate effective strategies to jointly learn multi-domain and multilingual NMT models under the incomplete data condition.
- Our empirical results show that MDML-NMT model can improve translation quality in the zero-shot directions by mitigating the **off-target translation** issue that an MNMT model translates the input sentence to a wrong target language. Additionally, MDML-NMT exhibits domain transfer ability by achieving up to +4 BLEU improvement over the multi-domain NMT on the translation direction where in-domain training data is absent. Thanks to the effective cross-domain and cross-lingual knowledge sharing, MDML-NMT outperforms the adapter-based method (Cooper Stickland et al., 2021) by a large margin in the language-domain zero-shot setting.
- Our study sheds light on effective MDML-NMT training. Our experimental results reveal that: (i) for the domain, it is important to make the encoder domain-aware by either providing the domain tags or training with the auxiliary task; and (ii) for the language, the best practice is to prepend the target language tag to the encoder.

2 Multi-domain Multilingual NMT

In this section, we first provide the necessary background on multilingual NMT (MNMT) and multi-domain NMT individually. We then describe effective modelling approaches for the integration of multi-domain and multilingual NMT (MDML-NMT).

2.1 Multilingual NMT

Given a set of languages L, the primary goal of MNMT is to learn a single NMT model that can handle all translation directions of interest in this set of languages (Dabre et al., 2020). According to the parameter sharing strategy, MNMT can be categorised into: 1) partial parameter sharing (Dong et al., 2015; Firat et al., 2016; Zhang et al., 2021), and 2) full parameter sharing (Ha et al., 2016; Johnson et al., 2017). The latter has been widely adopted because of its simplicity, lightweight, and its zero-shot capability. Thus, we adopt the full parameter sharing strategy in our work.

In the fully parameter-shared MNMT, all parameters of encoders, decoders and attentions are shared across tasks. Special language tags are introduced to indicate the target languages. One can prepend the target language tags to either the source or target sentences. The model is then trained jointly to minimise the negative log-likelihood across all training instances:

$$\mathcal{L}_{\text{ML}}(\boldsymbol{\theta}) := -\sum_{(s,t)\in T}\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{C}_{s,t}} \log P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}) \quad (1)$$

where $\boldsymbol{\theta}$ is model parameters, $\mathcal{C}_{s,t}$ denotes a bilingual corpus for the source language s and the target language t, $(\boldsymbol{x},\boldsymbol{y})$ is a pair of parallel sentences in the source and target language, and T denotes the translation tasks for which we have bitext available. Among all possible language pairs $(s,t) \in L \times L$, we often only have access to bilingual data for a subset of them. We denote these pairs as *seen* (observed) translation tasks, and the rest as *unseen* tasks corresponding to the zero-shot setting.

2.2 Multi-domain NMT

Multi-domain NMT aims to handle translation tasks across multiple domains for a given language pair. Similar to MNMT, tagging the training corpus is the most popular approach, where a tag indicates the domain of a sentence pair. We also minimise the negative log-likelihood across all domains to train the model:

$$\mathcal{L}_{\text{MD}}(\boldsymbol{\theta}) := -\sum_{d\in D}\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{C}^{d}_{s,t}} \log P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}) \quad (2)$$

where D is the set of domains, and $\mathcal{C}^d_{s,t}$ denotes the parallel bitext in the source language s, target language t, and the domain d.

Apart from tagging, some auxiliary tasks have also been incorporated into the training process. A common practice is the use of domain discrimination, which aims to force the encoder to capture *domain-aware* characteristics (Britz et al., 2017). For this purpose, a domain discriminator is added to the NMT model at training time. The input to the discriminator is the encoder output, and its output predicts the probability of the domain of the source sentence. The discriminator is jointly trained with the NMT model, and is discarded at inference time.

Let $\mathbf{h} = \text{enc}(\boldsymbol{x})$ be the representation of sentence $\boldsymbol{x}$ computed by the mean-pooling over the hidden states of the top layer of the encoder. The training objective for the domain-aware encoder is as follows:

$$\mathcal{L}_{\text{disc}}(\boldsymbol{\theta},\boldsymbol{\psi}) := -\sum_{d\in D}\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{C}^{d}_{s,t}} \log \Pr(d|\mathbf{h};\boldsymbol{\psi}) \quad (3)$$

$$\mathcal{L}_{\text{MD-aware}}(\boldsymbol{\theta},\boldsymbol{\psi}) := \mathcal{L}_{\text{MD}}(\boldsymbol{\theta}) + \lambda\mathcal{L}_{\text{disc}}(\boldsymbol{\theta},\boldsymbol{\psi}) \quad (4)$$

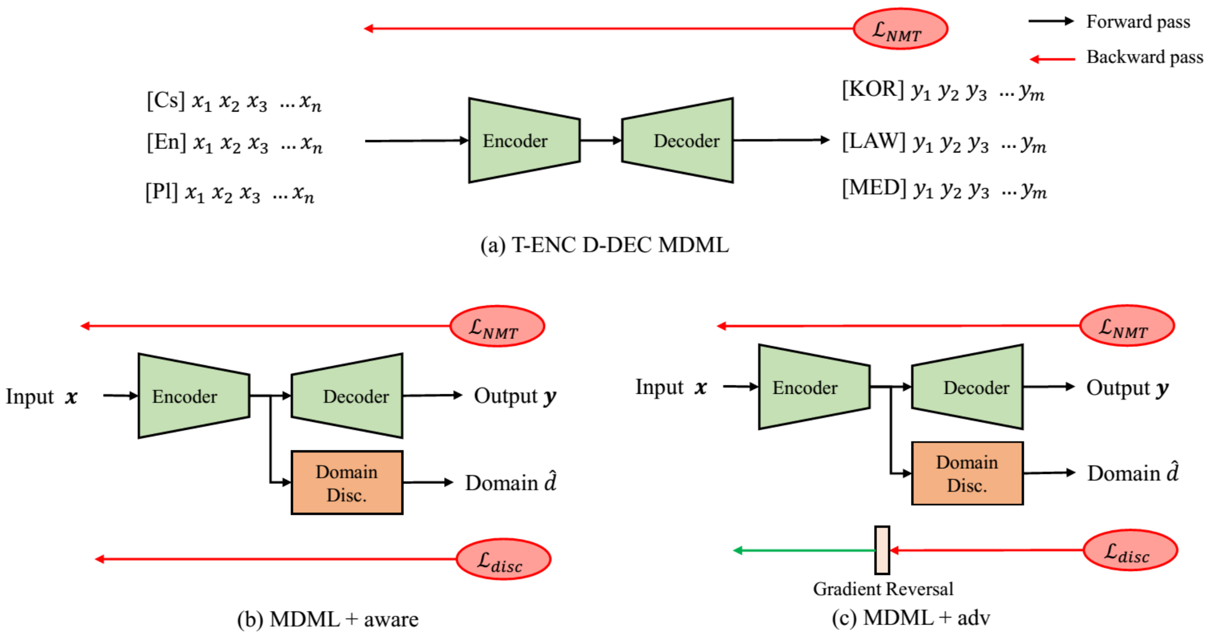
where $\boldsymbol{\psi}$ is the parameter of the domain discriminator classifier, and λ controls the contribution of the domain discriminator into the training objective of the multi-domain NMT model.

Alternatively, one can design an adversarial training objective in order to learn domain-agnostic representations by the encoder. This is achieved by inserting a gradient reversal layer (Ganin and Lempitsky, 2015) between the encoder and the domain discriminator. The gradient reversal layer behaves as an identity layer in the forward pass but reverses the gradient sign during back-propagation. It has the opposite effect on the encoder, forcing it to learn domain-agnostic representations. This encourages the domain specific characteristic to be learned mainly by the decoder of the NMT model.

2.3 Composition of Domains and Languages

In this paper, we explore strategies for composing multi-domain and multilingual NMT. We consider the incomplete multi-domain multilingual data condition where in-domain data may be only available in a subset of language pairs. For example, Table 1 shows one of the data conditions explored in our experiments in Section 3. Given the five language pairs and five domains, we assume that the domain data in some language pairs are missing. Our goal is to investigate effective techniques to train a high-quality MDML-NMT model covering all combinations of domains and language pairs.

Given a specific domain, we define *in-domain languages* as those having data available in the domain as part of some bilingual corpora; the rest



(a) T-ENC D-DEC MDML

(b) MDML + aware

(c) MDML + adv

Figure 2: Illustration of domain and languages composition strategies: (a) prepending domain (D) and target language (T) tag to encoder (ENC) or decoder (DEC). This example shows a T-ENC D-DEC model where the target language tag and domain tag are added to encoder and decoder respectively; (b) combining the tagging method with the domain aware auxiliary task (MDML + aware) to learn domain-aware representation; and (c) combining the tagging method with the domain adversarial auxiliary task (MDML + adv) to learn domain-agnostic representation.

Trans. direction	Eval. domain	MDML task type
En→De	LAW	seen in→in
En→Cs	LAW	seen in→out
Pl→En	LAW	seen out→in
De→Cs	LAW	unseen (zero-shot) in→out
Cs→De	LAW	unseen (zero-shot) out→in
Pl→Cs	LAW	unseen (zero-shot) out→out

Table 2: Examples of MDML task types in the leave-one-domain-out LAW training scenario of Table 1. Please refer to Table 1 for the in/out and seen/unseen settings.

of the languages are referred to as *out-of-domain languages*. We consider all combinations of in-domain/out-of-domain source/target languages for both seen and unseen translation directions (see examples in Table 2) in Section 3.

We investigate different combinations of the tagging strategy and auxiliary task training to effectively train MDML-NMT models, as shown in Figure 2.

Language and Domain Tags. We explore different ways of injecting the target language tags and domain tags into the translation process. Following the standard convention, we explore inserting the target language tag at the beginning of either the source sentence or the translation. Furthermore, the domain tag can also be added to either the source or the target side.

Auxiliary Task Training. We investigate the effect of encoder-based auxiliary tasks on MDML-NMT. As described in Section 2.2, we consider two types of auxiliary objectives to train encoder which are domain-aware or domain-agnostic. The former aims to amplify the domain-related features, while the latter focuses on the domain invariant representation in the encoder.

3 Experiments

In this section, we evaluate the MDML-NMT approaches and seek to answer the following research questions (**RQs**):

- **RQ1**: *Do out-of-domain translation tasks benefit from the out-of-domain and in-domain bitext in other translation pairs?*

  We explore the benefits of having a single MDML model trained on all available training data from multiple languages and domains over the multi-domain bilingual (MDBL) and the single domain multilingual (SDML) models learned on a subset of training data from a single language pair or domain. We carefully design controlled experiments to build incomplete data conditions and study the translation quality of the unified MDML-NMT model on both seen and unseen (zero-shot) translation directions. We hypothesise that the translation involving the out-of-domain languages can be beneficial from the in-domain languages thanks to the knowledge sharing across domain and languages.

- **RQ2**: *What is effective method to handle composition of domains and languages?*

  We investigate strategies for effective integration of existing multi-domain and multilingual NMT methods, including the use of language and domain tags and auxiliary task training.

3.1 Setup

We describe the experimental setup in this section, and then present our results.

Dataset. We conduct experiments with translation directions among five languages English (En), Czech (Cs), German (De), French (Fr) and Polish (Pl). Following the recipe in Koehn and Knowles (2017), we create five domains: Law (LAW) , IT (IT), Koran (KOR), Medical (MED), and Subtitles (SUB) from OPUS (Tiedemann, 2012). These corpora are deduplicated and randomly selected, from each corpus 2K sentences extracted as the development and test sets in all possible translation pairs. The statistics of the training dataset are reported in Appendix A.

Seen vs Unseen Language Pairs. We categorise the evaluated languages into two groups, high-resource languages including En, De, and Fr, for which bilingual data among these languages is easy to obtain. We also consider low-resource languages, including Cs and Pl, for which only English-centric data is available, resulting in two language pairs. As a result, there are five *seen* language pairs, consisting of ten seen translation directions.[1] There are also five *unseen* language pairs, resulting in ten unseen translation directions; they are the ones for which we do not have any bitext in the dataset.[2]

Leave-one-domain-out (LODO). We curate the incomplete MDML data condition by removing the data of one domain for the translations tasks involving low-resource languages. An example of the leave-one-domain-out data condition is shown in Table 1. In total, there are five LODO conditions, each of which corresponding to removing the bitext of one domain for both En-Cs and En-Pl (*i.e.,* our low-resource language pairs). For each of these LODO conditions, we have five seen language-pairs and five unseen language-pairs, hence a total of 20 translation tasks in both directions.

In the multi-domain NMT literature, this setting is related to domain generalisation which evaluates the NMT model on out-of-domain data in a zero-shot manner. By carefully removing only a specific domain, we would like to examine whether extra data (*i.e.,* the in-domain and out-of-domain data for high-resource languages, and out-of-domain data for low-resource languages) can boost the generalisation of MDML-NMT to the domain of interest.

Models. We use Transformer (Vaswani et al., 2017) as the NMT model architecture and Fairseq implementation (Ott et al., 2019). For all MDML-NMT models, we initialise them with mBART_large (Liu et al., 2020). We describe the model training details in Appendix B.

As described in Section 2.3, our approaches to MDML problem include combining language and domain tags, and adding domain auxiliary task to the standard multilingual NMT objective. In the first approach, the target language tags can be inserted to the source sentence (T-ENC) or the target sentence (T-DEC). The domain tags can also be handled in similar manners denoted as D-ENC and D-DEC respectively. On combining these tags, the language tag always appears first in the sentence. In addition to the domain and language tag combination, we also explore whether learning domain-aware or domain-agnostic representation in the encoder with auxiliary task can aid MDML-NMT performance. Figure 2 summarises the MDML-NMT approaches evaluated in this paper.

We also report the results of the adapter-based domain-specific MNMT, proposed by Cooper Stickland et al. (2021). Language adapters (Bapna and Firat, 2019) are firstly injected to each layer of a pre-trained MNMT model and then trained while freezing the backbone. Then, domain adapters are stacked on top of the language adapters and trained without backpropagating to the MNMT backbone and the language adapters. Since we do not consider any additional parallel

[1] This set consists of En-Fr, En-De, De-Fr, En-Cs, En-Pl.

[2] This set consists of De-Cs, De-Pl, Fr-Cs, Fr-Pl, Cs-Pl.

	D-Enc	D-Dec
MDBL	12.43	10.21
+adv	12.91	9.90
+aware	13.13	10.13

	D-Enc		D-Dec	
	T-Enc	T-Dec	T-Enc	T-Dec
MDML	14.48	13.21	14.11	8.16
+adv	14.91	14.30	14.72	8.44
+aware	15.00	14.59	**15.35**	7.99

Table 3: Average BLEU score of En→Cs translation across all leave-out domains for multi-domain multilingual (MDML) models and multi-domain bilingual (MDBL) models. The best score on overall and within each tagging group are marked in **bold** and underline respectively.

		seen -both	unseen -SDML	unseen -both
T-Enc	SDML	**41.40**	6.80	7.73
	MDML	37.25	**21.72**	**9.27**
T-Dec	SDML	**41.03**	7.79	8.16
	MDML	35.44	**21.43**	**14.73**

Table 4: Average BLEU scores of single-domain multilingual (SDML) and multi-domain multilingual (MDML) on the leave-out domains for three groups: (i) *seen-both* - the three seen high-resource language pairs (En-De, En-Fr, De-Fr); (ii) *unseen-SDML* - the two low-resource language pairs which are seen by MDML but unseen to SDML (En-Cs, En-Pl); and (iii) *unseen-both* - the other five unseen language pairs.

	MDML	+adv	+aware
T-Enc	25.17	28.91	**30.14**
T-Enc D-Enc	24.36	28.90	29.23
T-Enc D-Dec	22.10	29.43	29.94
T-Dec	24.82	29.14	29.52
T-Dec D-Enc	24.95	28.56	29.01
T-Dec D-Dec	19.19	17.68	14.37
Adapter-based		23.26	

Table 5: Average BLEU score of MDML-NMT models across all five leave-one-out scenarios. The best score overall and within each tagging group are marked in **bold** and underline respectively.

data apart from the multi-domain dataset, we train the MNMT backbone as well as the language and domain adapters using this multi-domain multilingual dataset (instead of Paracrawl) for fair comparison.

Evaluation. We report the detokenised BLEU scores calculated by SacreBLEU (Post, 2018) (Post, 2018) and the micro-average of BLEU score in a group as the measure of overall performance.[3]

3.2 Results and Discussions

Can multilinguality help the multi-domain learning? (MDBL vs. MDML) We first examine the potential of MDML over the counterpart multi-domain NMT model. Table 3 shows the BLEU scores of MDBL and MDML for En→Cs translation on various LODO settings. A breakdown of BLEU scores on leave-out domains is shown in Table 11 in the Appendix C. The MDBL models are trained on all En→Cs bilingual data except of the domain of interest. Within the same tagging method, augmenting the NMT training with the domain auxiliary objectives (*i.e.,* domain-aware and domain-agnostic encoders) enhances the translation performance. The MDML models consistently surpass the corresponding MDBL settings, with an exceptional case, where both domain and language tags are applied to the decoder (*i.e.,* T-Dec D-Dec). This observation suggests there is knowledge sharing from in-domain languages to out-of-domain languages.

Can multi-domain data help multilingual NMT? (SDML vs. MDML) SDML models are domain-specific multilingual NMT models trained on the multilingual dataset in a given domain. As in-domain parallel data is absent for several language pairs, the MDML models are exposed to more seen translation tasks than SDML models thanks to the availability of out-of-domain data. Hence, for a given domain, we divide the evaluation translation tasks into three groups: seen-both, unseen-SDML and unseen-both. The seen-both and unseen-both groups consist of translation directions which are observed and unobserved respectively by both models in training. The unseen-SDML group corresponds to those unseen by SDML, but seen by MDML models. We report the average performance of the MDML and SDML model on the

[3] nrefs:1|case:mixed|eff:no|tok:none|smooth:exp|version:2.0.0

		seen (10)			unseen (zero-shot) (10)			AVG
		in→in (6)	in→out (2)	out→in (2)	in→out (4)	out→in (4)	out→out (2)	
Adapter-based		34.32	11.76	**33.34**	7.38	6.86	6.84	16.75
T-Enc	MDML	37.25	**14.63**	29.05	7.93	10.35	9.79	18.17
	+adv	36.81	13.88	28.33	**10.91**	22.05	11.38	20.56
	+aware	37.50	14.31	29.09	10.61	24.50	11.94	21.33
T-Enc D-Enc	MDML	32.32	11.52	24.23	7.22	17.25	7.85	16.73
	+adv	37.24	13.66	31.17	10.20	24.21	11.67	**21.36**
	+aware	**37.57**	13.15	31.15	8.65	25.20	11.24	21.16
T-Enc D-Dec	MDML	31.94	10.55	22.14	5.88	8.63	5.83	14.16
	+adv	36.70	12.85	25.38	10.61	22.57	9.52	19.61
	+aware	37.47	12.08	25.59	10.08	22.41	9.01	19.44
T-Dec	MDML	31.44	11.25	23.62	7.63	20.39	8.59	17.15
	+adv	36.92	13.94	28.83	8.93	24.48	12.14	20.87
	+aware	37.20	14.00	28.62	10.30	23.95	**12.18**	21.04
T-Dec D-Enc	MDML	31.80	10.40	22.13	5.97	18.81	7.47	16.10
	+adv	36.35	13.22	27.96	8.46	24.35	10.21	20.09
	+aware	37.00	13.32	29.34	9.57	**25.89**	11.43	21.09
T-Dec D-Dec	MDML	30.17	4.77	24.72	3.65	14.08	4.43	13.64
	+adv	25.18	6.04	25.94	5.88	14.72	6.37	14.02
	+aware	20.61	5.50	23.27	5.72	7.64	6.40	11.52

Table 6: Average BLEU score on leave-out domain for different translation tasks. We categorise 20 translation direction into *seen* where the training data for the translation direction is available, otherwise *unseen*. *in* and *out* show whether the corresponding domain is observed during training or not (see Table 2 for a concrete example). The number in parentheses shows how many translation directions are in the corresponding category. The best score of each column overall and within each tagging group are marked in **bold** and underline respectively.

leave-out domains in Table 4. The detailed results on each leave-out domain can be found in Table 12 in the Appendix C. As expected, SDML works well on the seen directions (seen-both) but behaves badly on the zero-shot settings (unseen-SDML and unseen-both). We speculate it is due to the negative inference among domains. On the other hand, MDML outperforms SDML in unseen-SDML by a large margin thanks to the out-of-domain parallel data. Additionally, leveraging multi-domain data also helps to improve multilingual NMT on unseen-both tasks up to +6 BLEU score on average.

What is an effective method to MDML? We have previously shown the benefits of MDML over multi-domain and multilingual NMT models. The remaining question is how to integrate the multi-domain and multilingual approaches effectively. We report the average BLEU scores of different MDML methods across all five LODO scenarios and 20 translation tasks in Table 5. Similar to the previous observation on En→Cs translation, models with domain discriminator outperform the vanilla MNMT model in all tagging methods. More specifically, the domain-aware MNMT models (+aware) are the winning method in most scenarios. These results emphasise the importance of having domain-aware representation in the encoder. Furthermore, it shows MDML is more effective than the adapter-based approach.

As illustrated in Table 2, translation tasks in MDML setting can be categorised into seen and unseen (zero-shot) tasks involving the in-domain or out-of-domain languages. Table 6 reports the performance of MDML-NMT models in the leave-out domains on different task categories, e.g. Law in the example in Table 1. The results for other domains, i.e. excluding the leave-out domains, can be found in Appendix C. Consistent with previous findings, the domain discriminative mixing methods outperform the other models. While the best multilingual NMT model (MDML T-Enc) performs comparably with other MDML-NMT models on seen translation tasks, the main benefit of MDML-NMT models comes from unseen translation tasks. As expected, for both seen and unseen tasks, the quality of translation when translating into in-domain languages is consistently higher than into out-of-domain languages. Stacking the

		seen					unseen (zeroshot)			
		En	De	Fr	Cs	Pl	De	Fr	Cs	Pl
T-Enc	MDML	94.72	95.99	95.54	92.10	94.50	48.66	49.38	32.73	40.78
	+adv	94.81	96.01	95.33	91.62	95.06	75.56	85.93	59.18	66.57
	+aware	94.85	96.09	95.56	91.60	94.69	80.92	90.86	64.77	74.67
T-Enc D-Enc	MDML	92.55	95.54	95.06	91.21	94.12	73.99	72.85	44.69	58.83
	+adv	94.65	96.11	95.42	90.51	93.60	80.30	81.16	59.13	67.17
	+aware	94.67	96.18	95.44	90.35	92.69	81.21	84.05	61.10	66.92
T-Enc D-Dec	MDML	94.33	95.22	95.10	91.26	94.98	43.53	46.94	36.10	44.04
	+adv	94.86	95.81	95.44	91.55	94.64	87.23	90.67	69.01	74.42
	+aware	95.01	96.01	95.49	91.34	94.46	82.00	91.38	68.75	75.69
T-Dec	MDML	94.03	95.32	95.04	90.70	93.83	90.44	92.74	60.44	70.93
	+adv	94.68	96.05	95.44	91.72	94.84	86.03	88.64	52.45	64.01
	+aware	94.72	96.21	95.50	92.22	95.13	77.20	87.61	58.17	70.86
T-Dec D-Enc	MDML	92.72	95.51	95.06	89.72	92.23	85.75	89.82	56.74	68.56
	+adv	93.82	96.14	95.53	91.41	94.27	84.70	87.99	51.59	63.66
	+aware	94.19	96.12	95.54	91.54	93.80	79.93	87.00	60.22	72.77
T-Dec D-Dec	MDML	93.44	90.30	93.44	74.49	83.06	64.33	58.96	21.42	25.74
	+adv	80.29	17.71	94.36	49.29	16.07	3.37	47.72	1.04	0.62
	+aware	69.89	14.25	85.62	52.34	10.10	2.89	9.52	2.07	0.28

0	25	50	75	100

Table 7: On-target translation ratio of MDML-NMT models on the seen and unseen translation tasks.

language and domain adapters works particularly well in seen translation direction to in-domain target languages. Aligned with previous findings, the adapter-based method struggles to translate to out-domain target languages due to the unobserved combination of language and domain adapters during training (Cooper Stickland et al., 2021).

4 Analysis

4.1 Domain-specific token generation

In this section, we will look at how well MDML models are in generating domain-specific tokens. We concatenate all training data in a given domain in each language, remove stopwords, and extract the top 1000 domain-specific tokens with TF-IDF. The stopwords for each language are obtained from stopwords-iso[4]. Table 8 reports the F1 score of MDML models in generating leave-out domain-specific tokens. As expected, translation to in-domain languages (in→in, out→in) has a higher F1 score than translation to out-of-domain languages (in→out, out→out). Compared to MDML, both MDML-aware and MDML-adv models are able to generate more domain-specific tokens.

4.2 On-target translation ratio

One challenge of multilingual NMT (MNMT) is the off-target translation in zero-shot direction. Off-target translation is an issue that the MNMT model translates the input sentence to the wrong language, causing low BLEU scores. In this section, we assess the ability to alleviate the off-target issue in MDML models. Table 7 reports the on-target translation ratio of MDML models on seen and unseen translation for different target languages. We detect the language of translated targets using langdetect[5] tool and calculate the on-target translation ratio as the percentage of translated sentences having the target language detected correctly. As expected,

		in→in	in→out	out→in	out→out
T-Enc	MDML	63.22	21.45	35.58	16.42
	+adv	62.71	26.73	44.48	22.06
	+aware	63.45	25.85	47.53	23.96
T-Enc D-Enc	MDML	58.93	20.58	35.17	16.10
	+adv	63.14	24.24	46.75	23.55
	+aware	63.48	20.80	47.82	23.17
T-Enc D-Dec	MDML	58.82	20.32	30.34	13.37
	+adv	62.83	27.68	47.02	26.21
	+aware	63.59	27.64	47.21	25.73
T-Dec	MDML	58.35	21.70	43.69	19.98
	+adv	62.83	23.72	47.55	25.56
	+aware	63.08	25.90	47.31	25.49
T-Dec D-Enc	MDML	58.94	18.34	38.85	17.56
	+adv	62.37	21.86	45.67	20.31
	+aware	62.98	24.23	47.87	24.17
T-Dec D-Dec	MDML	56.74	12.52	40.01	10.98
	+adv	46.71	9.18	34.73	8.34
	+aware	39.20	8.93	26.32	8.06

0	25	50	75	100

Table 8: In-domain token generation F1 score.

[4] https://github.com/stopwords-iso/stopwords-iso

[5] https://github.com/Mimino666/langdetect

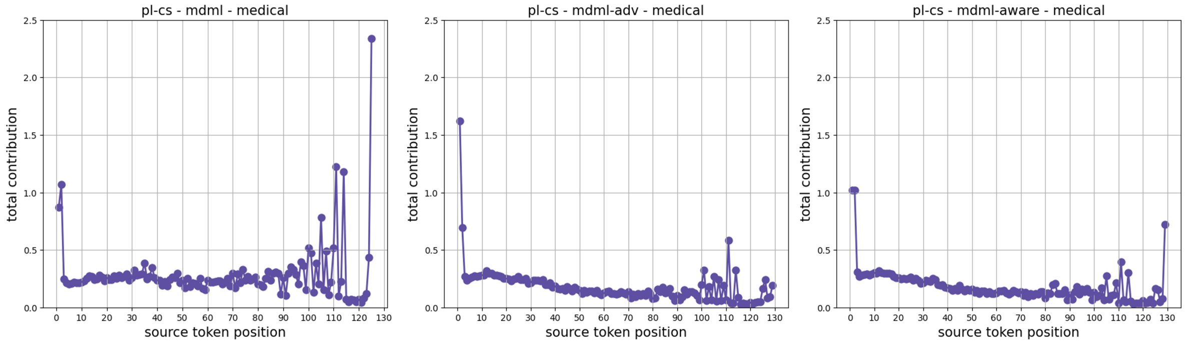


Figure 3: Source token contribution on Pl→Cs MDML with T-ENC D-ENC. The target language and domain tag are the first two tokens.

		En	De	Fr	Cs	Pl
LO	En		91.60	93.27	33.30	43.99
	De	93.26		90.79	9.88	10.82
	Fr	90.59	83.86		2.53	4.90
	Cs	91.51	68.39	64.78		7.87
	Pl	92.18	66.58	64.96	18.83	
others	En		95.38	94.79	84.79	92.83
	De	94.58		92.78	37.27	46.95
	Fr	91.70	86.50		22.04	28.20
	Cs	94.37	63.79	58.49		15.48
	Pl	94.66	63.29	56.46	13.21	
		0	25	50	75	100

Table 9: On target ratio of T-DEC D-DEC MDML on the leave-out (LO) and other domains. Rows and columns correspond to the source and target languages.

the seen translation tasks have more than 90% sentences in the correct target language, except T-DEC D-DEC models. On the other hand, the unseen tasks suffer from a low ratio, especially for Cs and Pl. We also observe significant improvement from MDML-aware and MDML-adv over the MDML models on unseen translation tasks to Cs and Pl.

Generally, T-DEC D-DEC model always underperforms other models and have a much lower on-target ratio on unseen tasks. Table 9 further confirms this phenomenon on the leave-out domains. While heavily suffering from the off-target issue in the leave-out domains, it has comparable ratios to other methods in other domains on seen tasks En-Pl and En-Cs. One possible explanation is that the combination of the target language and domain tags has never been observed during training for the unseen tasks with out-of-domain languages.

4.3 Language and domain tag contribution

To understand the role of the target and language tags to the generated prediction, we estimate the total contribution of source tokens at each position to the whole target sentence using Layerwise Relevance Propagation (Voita et al., 2021). We filter out the pairs having too short or too long target sentences and compute the contribution to target sentences of length between 10 and 100.

Results of T-ENC D-ENC MDML models on Pl→Cs translation in the leave-out medical domain are shown in Figure 3. The language and domain tag are the first two source tokens in respective order. It can be seen that all models have a similar trend in which the contribution of source tokens decreases toward later positions and suddenly increases at a few last positions. Additionally, the target language tags play an important role in the final prediction of all MDML models. Interestingly, while still having a fairly high contribution compared to other tokens, the domain tag seems less important for the domain adversarial models. It can be explained that the encoder learns to produce domain agnostic representation; hence less depends on the domain tags.

5 Related works

Multilingual NMT. As a remarkable branch of NMT, multilingual NMT (MNMT) has been appealing for its capability of supporting translations among different language pairs. Dong et al. (2015) opened the door to the MNMT by conducting a one-to-many translation. Firat et al. (2016) effectively extend this approach to a many-to-many setting. Since these approaches consider each translation as an independent system, they suffer from two major drawbacks. First, as the parameter size is proportional to the language size, it is not parameter-efficient when scaling to tens or hundreds of languages. In addition, the separate architectures cannot fully benefit from cross-lingual knowledge transfer. Johnson et al. (2017); Ha et al.

(2016) devise a universal MNMT system to alleviate these issues by prepending a target language tag to the inputs and training a shared SEQ2SEQ model on the concatenation of all bitext. However, owing to the negative interference, high-resource languages suffer from translation inferiority, compared to the corresponding bilingual NMT models. As a remedy, Zhang et al. (2021); Kudugunta et al. (2021) leverage a mixture-of-experts design to separate language-specific features from the generic features by incorporating language-specific components into the universal MNMT model. Besides, Bapna and Firat (2019); Zhu et al. (2021) propose to fine-tune a lightweight adapter as a means of compensation for the quality loss caused by the adverse effect.

Multidomain NMT. While both involving training on dataset coming from multiple domains, NMT domain adaption is different from multi-domain NMT. The former aims to transfer the knowledge of out-of-domain data into the in-domain data (Luong and Manning, 2015; Zoph et al., 2016; Freitag and Al-Onaizan, 2016), while the latter focuses on building a system, performing well on multiple domains (Pham et al., 2021). Since lexical and topic variations have been observed in different domains, it is challenging to handle the mixed-domain data with a generic NMT model (Farajian et al., 2017). To operate translation in multiple domains, recent research focuses on exploiting domain-shared and domain-specific knowledge by introducing a domain tag to the source sentence (Kobus et al., 2017), using auxiliary objectives such as domain discrimination loss (Britz et al., 2017; Gu et al., 2019), domain knowledge distillation (Currey et al., 2020), and modifying the architecture to capture this information explicitly (Zeng et al., 2018). Rather than using a heavy domain-specific encoder-decoder architecture, Wang et al. (2020) introduce lightweight domain transformation layers between the shared encoder and decoder.

Multilingual & multi-domain NMT. Previous works have mainly considered multilingual and multi-domain NMT models as two disjoint systems. Until recently, Cooper Stickland et al. (2021) propose to unify these two settings into a holistic system, but focus more on the domain adaptation angle. They investigate the combination of language and domain adapters by superimposing domain adapters on language adapters. They noticed that domain adapters and back-translation could boost the translation quality on the out-of-domain languages. In contrast, our work creatively stitches multilingual and multi-domain NMT together and explores the capability of a cross-lingual domain transfer within a unified model without adaption.

6 Conclusion

We study the problem of MDML-NMT for which a single NMT can support multiple translation directions and domains. We investigate whether the tagging and auxiliary task learning method can be combined for MDML-NMT. Our empirical results reveal a positive transfer from in-domain to out-of-domain languages, especially in the zero-shot scenario. This study provides insights into the synergy of the domain and language aspects of training an MDML-NMT model. The main findings include: (i) it is crucial to make the encoder domain-aware; and (ii) it is best to prepend the target language tag to the encoder in MDML. These findings lay the groundwork for future research in this direction.

Acknowledgments

This research is supported by an eBay Research Award and the ARC Future Fellowship FT190100039. This work is partly sponsored by the Air Force Research Laboratory and DARPA under agreement number FA8750-19-2-0501. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The authors are grateful to Minghao Wu, Michelle Zhao and the anonymous reviewers for their helpful comments to improve the manuscript.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Alexandre Bérard, Zae Myung Kim, Vassilina Nikoulina, Eunjeong Lucy Park, and Matthias Gallé. 2020. A multilingual neural machine translation model for biomedical data. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, Online. Association for Computational Linguistics.

Denny Britz, Quoc Le, and Reid Pryzant. 2017. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 118–126, Copenhagen, Denmark. Association for Computational Linguistics.

Asa Cooper Stickland, Alexandre Berard, and Vassilina Nikoulina. 2021. Multilingual domain adaptation for NMT: Decoupling language and domain information with adapters. In *Proceedings of the Sixth Conference on Machine Translation*, pages 578–598, Online. Association for Computational Linguistics.

Anna Currey, Prashant Mathur, and Georgiana Dinu. 2020. Distilling multiple domains for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4500–4511, Online. Association for Computational Linguistics.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Computing Surveys (CSUR)*, 53(5):1–38.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

M. Amin Farajian, Marco Turchi, Matteo Negri, Nicola Bertoldi, and Marcello Federico. 2017. Neural vs. phrase-based machine translation in a multi-domain scenario. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 280–284, Valencia, Spain. Association for Computational Linguistics.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897*.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR.

Shuhao Gu, Yang Feng, and Qun Liu. 2019. Improving domain adaptation translation with domain invariant and specific information. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3081–3091, Minneapolis, Minnesota. Association for Computational Linguistics.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798*.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378, Varna, Bulgaria. INCOMA Ltd.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Minh-Thang Luong, and Orhan Firat. 2021. Beyond distillation: Task-level mixture-of-experts for efficient inference. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3577–3599.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Minh-Thang Luong and Christopher Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 76–79, Da Nang, Vietnam.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

MinhQuang Pham, Josep Maria Crego, and François Yvon. 2021. Revisiting multi-domain machine translation. *Transactions of the Association for Computational Linguistics*, 9:17–35.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of LREC*.

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. Facebook ai's wmt21 news translation task submission. In *Proc. of WMT*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Elena Voita, Rico Sennrich, and Ivan Titov. 2021. Analyzing the source and target contributions to predictions in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

Yong Wang, Longyue Wang, Shuming Shi, Victor OK Li, and Zhaopeng Tu. 2020. Go from the general to the particular: Multi-domain translation with domain transformation networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9233–9241.

Jiali Zeng, Jinsong Su, Huating Wen, Yang Liu, Jun Xie, Yongjing Yin, and Jianqiang Zhao. 2018. Multi-domain neural machine translation with word-level domain context discrimination. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 447–457, Brussels, Belgium. Association for Computational Linguistics.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021. Share or not? learning to schedule language-specific capacity for multilingual translation. In *International Conference on Learning Representations*.

Yaoming Zhu, Jiangtao Feng, Chengqi Zhao, Mingxuan Wang, and Lei Li. 2021. Serial or parallel? plug-able adapter for multilingual machine translation. *arXiv preprint arXiv:2104.08154*.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575.

Domain	Cs-En	De-En	Fr-En	Pl-En	De-Fr
LAW	1.3M	467K	596K	1.0M	1.3M
IT	73K	158K	230K	97K	146K
MED	686K	705K	705K	666K	707K
KOR	117K	17.8K	28K	30K	10K
SUB	595K	494K	492K	491K	590K

Table 10: Number of training sentences in the evaluation datasets. Each dataset contains 2K dev and test sentences.

A Data statistics

Table 10 shows the statistics of dataset used in the experiments.

B Training Details

For all MDML-NMT models, we initialise them with mBART_large (Liu et al., 2020) and train with mixed-precision training up to 200K update steps (around 13 epochs) using a batch size of 8192 tokens and early stopping on 8 V100 GPUs. The multi-domain NMT (MDBL) is trained in a similar manner, except with the total update steps of 60K which is equivalent to around 30 epochs. We apply Adam with an inverse square root schedule, a linear warmup of 5000 steps and a learning rate of 3e-5. We set dropout and label smoothing with a rate of 0.3 and 0.2. We use temperature-based sampling with $T = 5$ to balance training size between domains and languages (Arivazhagan et al., 2019).

For the NMT model with auxiliary task, the domain discriminator is a 2-layer feed-forward network with hidden size of 1024. We set the mixing hyperparameters λ in Equation 4 to 1, *i.e.,* the domain discriminative loss and NMT loss contributes equally to the training signal.

Followed (Cooper Stickland et al., 2021), we use adapter bottle-neck of 1024 for the adapter-based models. The monolingual language adapters are trained all together on the multi-domain dataset while the NMT backbone are frozen. In contrast, we train domain adapters separately for each domain and build homogeneous batches containing sentences from the same language direction and domain. We also apply domain-adapter dropout (DADrop) where the domain adapters are skipped 20% of time.

C Additional Results

MDBL vs. MDML. Table 11 shows the BLEU scores of different models for En→Cs translation on various LODO settings. Each domain column reports the results corresponding to the LODO setting in which the bitext of that domain is removed.

SDML vs. MDML. We report the performance of the MDML and SDML model on each leave-out domains in Table 12.

MDML Result. The average BLEU scores on each domain across all five LODO scenarios and 20 translation tasks are reported in Table 13. Table 14 reports the performance of MDML-NMT models on other domains (excluding the leave-out domains) on different task categories.

		LAW	IT	KOR	MED	SUB	AVG
D-ENC	MDBL	10.52	20.37	7.63	19.46	4.16	12.43
	+adv	10.62	19.43	8.16	20.79	5.57	12.91
	+aware	10.37	21.70	8.25	20.07	5.26	13.13
D-DEC	MDBL	9.21	12.39	6.64	19.56	3.27	10.21
	+adv	9.78	11.01	6.76	18.03	3.94	9.90
	+aware	9.51	12.25	7.00	18.46	3.41	10.13
T-ENC D-ENC	MDML	11.98	22.64	8.08	21.05	8.63	14.48
	+adv	12.11	**23.43**	**9.26**	21.59	8.16	14.91
	+aware	11.82	23.07	9.08	21.54	9.51	15.00
T-DEC D-ENC	MDML	10.57	18.32	6.87	20.04	10.25	13.21
	+adv	11.36	22.69	7.13	20.88	9.44	14.30
	+aware	11.25	21.94	8.73	20.69	10.34	14.59
T-DEC D-DEC	MDML	5.21	17.56	4.53	9.12	4.36	8.16
	+adv	2.25	17.47	4.89	12.41	5.18	8.44
	+aware	3.37	18.85	4.23	9.35	4.14	7.99
T-ENC D-DEC	MDML	9.39	21.29	8.28	22.06	9.54	14.11
	+adv	10.84	22.92	8.45	22.27	9.10	14.72
	+aware	**12.29**	22.70	8.39	**22.62**	**10.75**	**15.35**

Table 11: BLEU score of En→Cs translation on leave-out domains for multi-domain multilingual (MDML) models and multi-domain bilingual (MDBL) models. +adv and +aware denote MDML models trained with domain-agnostic or domain-aware auxiliary tasks, respectively. The best score on each domain overall and within each tagging group are marked in **bold** and underline respectively.

		LAW	IT	KOR	MED	SUB	AVG	
T-ENC	(I)	SDML	**49.21**	**41.63**	**32.33**	**51.84**	**32.01**	**41.40**
		MDML	45.87	35.76	29.01	47.30	28.30	37.25
	(II)	SDML	1.98	13.29	3.03	12.57	3.11	6.80
		MDML	**23.40**	**27.27**	**13.29**	**31.19**	**13.44**	**21.72**
	(III)	SDML	2.68	14.89	4.26	11.70	5.10	7.73
		MDML	**5.07**	**15.32**	**6.26**	**12.87**	**6.85**	**9.27**
T-DEC	(I)	SDML	**48.42**	**41.36**	**29.50**	**54.00**	**31.88**	**41.03**
		MDML	44.48	30.43	28.53	45.75	27.99	35.44
	(II)	SDML	2.07	14.01	3.95	14.54	4.36	7.79
		MDML	**21.73**	**28.84**	**13.77**	**29.18**	**13.65**	**21.43**
	(III)	SDML	2.66	15.37	4.38	12.94	5.44	8.16
		MDML	**14.57**	**15.57**	**14.39**	**20.52**	**8.61**	**14.73**

Table 12: Average BLEU scores of single-domain multilingual (SDML) and multi-domain multilingual (MDML) on the leave-out domains for three groups: (I) the three seen high-resource language pairs (En-De, En-Fr, De-Fr); (II) the two low-resource language pairs which are seen by MDML but unseen to SDML (En-Cs, En-Pl); and (III) the other five unseen language pairs.

	model	Law	IT	Kor	Med	Sub	AVG
Adapter-based		23.02	29.37	19.52	28.87	15.51	23.26
T-Enc	MDML	23.09	27.86	22.83	34.19	17.88	25.17
	+adv	28.74	31.14	25.68	40.08	18.89	28.91
	+aware	31.56	**32.00**	26.63	**40.88**	19.62	**30.14**
T-Enc D-Enc	MDML	21.14	26.92	21.84	34.61	17.31	24.36
	+adv	26.09	31.91	26.09	40.72	19.67	28.90
	+aware	27.10	31.85	26.40	40.77	**20.03**	29.23
T-Enc D-Dec	MDML	20.04	24.71	19.57	30.77	15.43	22.10
	+adv	30.14	31.26	26.03	41.04	18.68	29.43
	+aware	**31.61**	31.49	26.56	40.84	19.20	29.94
T-Dec	MDML	25.92	26.81	20.49	34.34	16.56	24.82
	+adv	29.64	30.91	26.22	40.31	18.62	29.14
	+aware	29.45	31.54	**26.81**	40.82	19.01	29.52
T-Dec D-Enc	MDML	24.89	27.23	20.97	34.83	16.85	24.95
	+adv	27.66	31.02	25.54	39.55	19.02	28.56
	+aware	28.15	31.31	25.77	40.27	19.56	29.01
T-Dec D-Dec	MDML	21.24	19.24	16.06	27.42	12.01	19.19
	+adv	20.58	20.60	11.75	24.09	11.40	17.68
	+aware	13.91	17.75	9.49	20.97	9.74	14.37

Table 13: Average BLEU score of MDML-NMT models on each domain across all five leave-one-out scenarios and 20 (seen and unseen) translation tasks. The best score on each domain overall and within each tagging group are marked in **bold** and underline respectively.

		seen (10)						unseen (zero-shot) (10)					
		Law	IT	Kor	Med	Sub	AVG	Law	IT	Kor	Med	Sub	AVG
Adapter-based		43.38	36.15	26.92	50.06	27.27	36.76	10.97	23.23	27.69	35.05	12.16	21.82
T-Enc	MDML	43.07	36.66	26.32	49.37	26.28	36.34	4.16	21.07	23.45	22.67	11.12	16.49
	+adv	42.83	35.73	26.60	49.04	25.46	35.93	17.32	28.73	28.58	35.38	14.17	24.84
	+aware	43.36	36.54	27.08	49.84	25.97	36.56	22.88	29.64	29.91	36.62	15.38	26.88
T-Enc D-Enc	MDML	36.02	36.50	22.29	43.00	22.48	32.06	7.01	26.46	22.79	28.34	12.28	19.38
	+adv	43.08	36.41	26.76	49.46	25.99	36.34	10.63	29.77	28.31	36.49	15.02	24.05
	+aware	43.41	36.71	27.02	49.74	26.15	36.61	12.95	29.20	29.25	36.64	15.37	24.68
T-Enc D-Dec	MDML	36.06	36.32	22.24	42.74	21.86	31.85	3.87	21.95	19.36	21.10	9.92	15.24
	+adv	42.80	35.67	26.66	49.10	25.23	35.89	20.29	29.92	29.45	37.91	14.79	26.47
	+aware	43.53	36.50	27.39	49.94	25.84	36.64	23.20	29.72	29.82	36.98	15.32	27.01
T-Dec	MDML	35.30	35.45	21.60	42.39	21.68	31.28	17.02	27.49	20.37	28.55	12.00	21.08
	+adv	42.76	35.95	26.90	49.36	25.56	36.10	18.73	28.04	29.25	36.22	13.33	25.11
	+aware	43.22	36.28	27.33	49.70	25.72	36.45	17.84	29.02	30.25	36.75	14.23	25.62
T-Dec D-Enc	MDML	35.94	35.74	21.64	42.31	22.11	31.55	17.00	28.23	21.68	29.42	11.60	21.59
	+adv	42.08	35.45	26.85	48.55	25.41	35.67	15.87	28.49	29.12	34.95	13.71	24.43
	+aware	43.12	35.62	25.98	48.69	25.90	35.86	16.02	28.99	28.40	36.31	14.62	24.87
T-Dec D-Dec	MDML	33.99	32.09	20.85	39.35	18.75	29.01	9.10	13.17	11.46	16.37	4.82	10.98
	+adv	35.67	27.11	19.13	35.53	17.76	27.04	6.92	15.26	4.74	14.39	5.75	9.41
	+aware	26.89	23.03	17.07	31.83	16.26	23.02	3.14	13.03	2.58	10.93	3.92	6.72

Table 14: Average BLEU score on other domains, i.e. excluding the leave-out domains, for different translation tasks. We categorise 20 translation direction into *seen* where the translation direction in which training data are available, otherwise *unseen*. The number in parentheses shows how many translation directions in the corresponding category.

DUTNLP Machine Translation System for WMT22 General MT Task

Ting Wang **Huan Liu** **Junpeng Liu** **Degen Huang**[*]
School of Computer Science, Dalian University of Technology
{Wting_1513577,liuhuan4221,liujunpeng_nlp}@mail.dlut.edu.cn
huangdg@dlut.edu.cn

Abstract

This paper describes DUTNLP Lab's submission to the WMT22 General MT Task on four translation directions: English to/from Chinese and English to/from Japanese under the constrained condition. Our primary system are built on several Transformer variants which employ wider FFN layer or deeper encoder layer. The bilingual data are filtered by detailed data pre-processing strategies and four data augmentation methods are combined to enlarge the training data with the provided monolingual data. Several common methods are also employed to further improve the model performance, such as fine-tuning, model ensemble and post-editing. As a result, our constrained systems achieve 29.01, 63.87, 41.84, and 24.82 BLEU scores on Chinese → English, English → Chinese, English → Japanese, and Japanese → English, respectively.

1 Introduction

DUTNLP Lab participates in the WMT22 General MT Task on four translation directions: English ↔ Chinese and English ↔ Japanese. Our translation system is trained on the officially provided bilingual and monolingual data under the constrained condition. Several strategies such as fine-grained data pre-processing, large-scale synthetic data augmentation, diverse model architectures and domain fine-tuning are utilized to enhance the performance of the final ensemble model.

Since the quality of the training data is crucial to the translation performance, all the training sets are filtered by the off-the-shelf toolkits and some manual rules. Details will be discussed in Section 2. Those data pre-processing strategies are also employed to filter out the synthetic data generated by different data augmentation methods.

To generate synthetic parallel data, four data augmentation methods including back-translation (Sennrich et al., 2016), forward-translation (Wu et al., 2019), knowledge distillation (Freitag et al., 2017) and R2L training (Liu et al., 2016) are employed in our experiments. Specifically, we leverage source-side monolingual data by exploring forward-translation, knowledge distillation and R2L training, while target-side monolingual data by back-translation. These strategies increase the data size to a large extent. The generated data and the original parallel data are combined to train NMT models.

For model architectures, starting from Transformer-Big (Vaswani et al., 2017) settings, several Transformer variants are used to improve the model capacity and diversity. Previous studies (Bapna et al., 2018; Li et al., 2020) have shown that the translation performance can be significantly improved by increasing the model capacity. Therefore, we build different model architectures with either wider FFN layers (Ng et al., 2019) or deeper transformer encoder (Sun et al., 2019). Moreover, the Pre-Norm (Wang et al., 2019) is also adopted in all our experiments as its performance and training stability are better than the Post-Norm counterpart.

Domain fine-tuning is the most effective method in our experiments, which greatly improves the translation performance. We first employ previous WMT test sets as the domain data to fine-tune several models with different architectures. Then we ensemble those fine-tuned models and translate the test sets to construct pseudo parallel data. Finally, the original and the pseudo test sets are merged for further domain fine-tuning.

This paper is structured as follows: Section 2 describes the data pre-processing strategies. We present the details of our systems in Section 3 and show the experiment settings and results in Section 4. We draw the conclusion in Section 5.

[*] Corresponding author

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 397 - 402
December 7-8, 2022 ©2022 Association for Computational Linguistics

2 Data Pre-processing

For each language pair, we follow the constrained data requirements and make full use of the provided bilingual and monolingual data. Table 1 lists the data we used in our experiments.

Language Pair	Filtered Bilingual	Monolingual
En-Zh	34.5M	En:15M Zh:15M
En-Ja	20.1M	En:20M Ja:20M

Table 1: Statistics of the training dataset.

As the quality of the parallel training data is crucial to the final translation performance, we perform fine-grained data filtering with the off-the-shelf toolkits and some manual rules. For both language pairs, the pre-processing strategies are as follows:

- Normalize punctuation with Moses scripts (Koehn et al., 2007) for English. Chinese and Japanese text are separately segmented by jieba[1] and MeCab[2] toolkits.
- Filter out the duplicated sentence pairs.
- Filter out sentences containing html tags, illegal characters and invisible characters.
- Filter out sentences with the character-to-word ratio higher than 12 or lower than 1.5 following (Wei et al., 2021).
- Filter out sentences with the source-to-target token ratio higher than 3 or lower than 0.3 following (Wei et al., 2021).
- Filter out sentences in other languages by applying language identification (Joulin et al., 2016).
- Filter out sentence pairs with low alignment score by using fast-align (Dyer et al., 2013).
- For Chinese, we convert full-width format to half-width format and convert traditional Chinese characters to simplified ones.

3 System Overview

3.1 Model Architectures

Previous studies (Bapna et al., 2018; Li et al., 2020; Wei et al., 2021; Li et al., 2021) have shown that the translation performance can be significantly improved by increasing the model capacity. Considering the model performance, we adopt the Deep Encoder and Shallow Decoder architecture with wider FFN layer. For En-Zh pair, we adopt the Deep 35-6 big model as baseline model following (Wei et al., 2021). For En-Ja pair, in view of the training cost we choose the Deep 24-6 big model as baseline model following (Subramanian et al., 2021; Zhou et al., 2021). The details about the models are as follows:

- **Deep 24-6 model**: This model features 24-layer encoder, 6-layer decoder, 512 dimensions of word vector, 4096 dimensions of FFN, 16-head self-attention and uses Pre-Norm strategy(Wang et al., 2019).
- **Deep 35-6 big model**: This model features 35-layer encoder, 6-layer decoder, 768 dimensions of word vector, 3076 dimensions of FFN, 16-head self-attention and uses Pre-Norm strategy(Wang et al., 2019).

3.2 Data Augmentation

In this task, four data augmentation strategies are utilized to generate synthetic data, which have shown their effectiveness on improving the performance of NMT model in previous works (Wei et al., 2021; Zhou et al., 2021; Wang et al., 2021; Zeng et al., 2021).

Back-Translation (Sennrich et al., 2016) is the most commonly used data augmentation technique which generates pseudo parallel data by translating the target monolingual sentences into source language with a pre-trained target-to-source NMT model. Our back-translation is divided into three stages:

- Training an ensemble target-to-source NMT model with the provided bilingual parallel data.
- Translating the target monolingual sentences to source language with the pre-trained target-to-source NMT model to generate synthetic parallel data.
- Training models with the bilingual and synthetic parallel data in a ratio of 1:1.

[1] https://github.com/fxsjy/jieba
[2] http://taku910.github.io/mecab/

Forward-Translation (Wu et al., 2019) is another data generation technique. Different from back-translation, forward-translation translating the source monolingual corpus into target corpus with a pre-trained source-to-target NMT model. Here, the forward-translation is only applied to Ja → En direction.

Knowledge Distillation (Freitag et al., 2017) is a powerful technique to improve a student model by distilling knowledge from a group of teacher models. In our experiments, we first train several teacher models on the original bilingual data and generate synthetic training corpus with the ensemble teacher models. Then the student model is trained on the combination of the original and synthetic training set.

R2L Training Previous work (Liu et al., 2016) has shown that R2L training is an effective way to boost translation quality by addressing the error propagation problem in auto-regressive generation tasks. Following this strategy, we train an R2L model with the original source sentences and inverse target sentences and translate the source monolingual sentences into target sentences. In our experiment, we mix the synthetic data generated by both R2L and L2R models to for iterative joint training.

3.3 Domain Fine-tuning

Domain fine-tuning plays a key role in improving the model performance. Following Sun et al. (2019), we take previous development and test sets as in-domain data and fine-tune the models. For En ↔ Ja task, since previous development and test sets are too small to use, we search for additional in-domain data which are similar to the development sets. Specifically, we obtain the low-frequency domain-specific words in the development/test sets by employing the TF-IDF algorithm and filter sentences in the training set which contain those words.

3.4 Model Ensemble

Model ensemble is a widely used method in previous WMT shared tasks (Garmash and Monz, 2016), which can enhance the translation performance by combining the predictions of several models at each decoding step. In our work, we employ two kinds of ensemble methods, namely, checkpoint average and voting based ensemble. For checkpoint average, we average the top-5 checkpoints of each model according to their BLEU performance on the development set. While for model ensemble, we train several models with different architectures to increase the model diversity.

3.5 Post-editing

We apply post-editing to obtain the final translation outputs. For En → {Zh, Ja}, the post-editing includes removing the redundant spaces, converting punctuation to the language-specific format and replacing some of the English in the translation (such as the person name) with the English in the source sentence. For {Ja, Zh} → English, we de-tokenize the sentences with the Moses toolkit.

4 Experiments and Results

4.1 Settings

The implementation of our models is based on open-source fairseq (Ott et al., 2019) and we use sacreBLEU (Post, 2018) to measure system performances which is officially recommended. We select Transformer-big as the baseline for all tasks. The Zh ↔ En models are carried out on single NVIDIA 3090 GPU which has 24GB of memory and the Ja ↔ En models are carried out on 8 RTXA6000 GPUs each of which has 48GB of memory. For all tasks, the dropout probabilities are set to 0.1. We use the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.997$ (Zhou et al., 2021) during training. Table 2 lists the fairseq parameter setting in training.

Parameter	**Zh ↔ En**	**Ja ↔ En**
batch size	4096	8192
update-freq	4	2
learning rate	0.0005	0.002
warmup steps	4000	8000
save-interval-updates	4000	2000

Table 2: Fairseq parameter setting in training.

4.2 Zh ↔ En

For Zh ↔ En tasks, the training data consists of ParaCrawl v9, News Commentary v16, Wiki Titles v3, WikiMatrix, UN Parallel Corpus V1.0 (Ziemski et al., 2016) and CCMT Corpus. We take newsdev2017 as the development set and newstest2021 as the test set to tune the hyper-parameters. The training data is filtered by aforementioned methods and obtain the training data of 34.5M. The joint

System	En → Zh	Zh → En
Baseline	32.1	23.4
+ Back Translation	32.3(+0.2)	23.5(+0.1)
+ Checkpoint Average	32.8(+0.5)	24.2(+0.7)
+ Domain Fine-tuning	34.1(+1.3)	26.9(+2.7)
+ Ensemble	34.5(+0.4)	27.4(+0.5)
+ Post-edit	37.9(+3.4)	27.4

Table 3: The experimental result of En ↔ Zh task.

vocabulary with 32K words is generated by using sentencepiece(Kudo and Richardson, 2018). The officially provided back-translation data are not used in our experiments since no obvious improvements are obtained when adding it to the training set. The results of En ↔ Zh on newstest2021 are shown in Table 3.

We perform back-translation with the deep 35-6 big model in the target-to-source direction to generate the synthetic parallel data. Comparing with the baseline model, the back-translation technique leads to an improvement of 0.2 and 0.1 BLEU in En → Zh and Zh → En directions, respectively. The checkpoint average method brings another BLEU improvements of 0.5 and 0.7.

In the fine-tuning stage, we use previous WMT test sets as the in-domain data. We first perform fine-tuning on several different models with the combination of newstest2017-2019. Then we translate the in-domain data by the ensemble model to obtain pseudo parallel data and perform further fine-tuning on both the original and pseudo data. In our final submission, we add the newstest2020 and newstest2021 test set to the in-domain data. Domain fine-tuning is the most effective method in our experiment, which achieve an improvement of 1.3 and 2.7 BLEU scores in En → Zh and Zh → En directions, respectively.

We ensemble several models with better performance on the test set, in order to obtain more robust translation system. In our work, model ensemble further lead to a 0.4 and 0.5 BLEU improvement, respectively. Moreover, we apply post-editing to the translation outputs. It should be noted that post-editing can mainly improve the BLEU of En → Zh, which is about 3.4 BLEU. The punctuation format of Chinese translation has a great impact on BLEU. Finally, we obtain 37.9 BLEU scores in En → Zh direction and 27.4 BLEU scores in Zh → En direction.

4.3 Ja ↔ En

For Ja ↔ En tasks, we choose ParaCrawl v9, News Commentary v16, Japanese-English Subtitle Corpus (Pryzant et al., 2018), The Kyoto Free Translation Task Corpus (Neubig, 2011) and TED Talks as the training bilingual corpus. The final training bilingual corpus we used to train the model is about 20.1M. The source and target side each has a vocabulary with 32K words. We use the combination of newsdev2020 and newstest2020 as the development set and newstest2021 as the test set, respectively. Table 4 summarizes our results on newstest2021.

As shown in Table 4, all the four data augmentation methods improve the translation performance in both translation directions. We apply the deep 24-6 model to implement four data augmentation methods and We find that back-translation contributes the largest BLEU improvements (+4.1 BLEU) of the four data augmentation methods on En → Ja direction, while knowledge distillation performs best in the opposite direction (+2.1 BLEU). Moreover, we also evaluate the combination of the four data augmentation methods. In En → Ja direction, we combine the synthetic data from back-translation and R2L model with the original parallel data in a ratio of 0.5:0.5:1. By contrast, in Ja → En direction, we mix the synthetic data from back-translation, forward-translation and knowledge distillation with the original parallel data in a ratio of 0.5:0.5:0.5:1. The combination of multiple data augmentation methods brings 4.5 and 2.0 BLEU gains in En → Ja and Ja → En directions.

We further use newsdev2020, newstest2020 and selected in-domain data to fine-tune the model and achieve another 3.6 and 1.8 BLEU improvement in En → Ja and Ja → En directions, respectively. Then, the model ensemble further bring 1.1 and 0.6 BLEU improvement. Finally, we apply post-editing to the translation outputs and it further bring 0.2 and 0.1 BLEU improvement in En → Ja and Ja → En directions.

5 Conclusion

This paper presents the DUTNLP Translation systems for WMT22 General MT Task. Our main exploration is to improve the translation performance with the fine-grained data filtering, diverse model architectures, large-scale data augmentation and domain fine-tuning. The effectiveness of each method is demonstrated in our experiments. Model

System	En → Ja	Ja → En
Baseline	36.8	22.3
+ Back Translation	40.9	23.8
+ Forward Translation	39.5	24.2
+ Knowledge Distillation	38.9	24.3
+ R2L Training	39.7	23.4
+ BT+R2L	41.3(+4.5)	-
+ BT+FT+KD	-	24.3(+2.0)
+ Domain Fine-tuning	44.9(+3.6)	26.1(+1.8)
+ Ensemble	46.0(+1.1)	26.7(+0.6)
+ Post-edit	46.2(+0.2)	26.8(+0.1)

Table 4: The experimental result of En ↔ Ja task.

ensemble and post-editing are also used to further improve the performance of our system. Our constrained systems achieve 29.01, 63.87, 41.84, and 24.82 BLEU scores on Chinese → English, English → Chinese, English → Japanese, and Japanese → English, respectively.

Acknowledgements

We sincerely thank the reviewers for their insightful comments and suggestions to improve the paper. The authors gratefully acknowledge the financial support provided by the National Key Research and Development Program of China (2020AAA0108004) and the National Natural Science Foundation of China under (No.U1936109).

References

Ankur Bapna, Mia Chen, Orhan Firat, Yuan Cao, and Yonghui Wu. 2018. Training deeper neural machine translation models with transparent attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3028–3033, Brussels, Belgium. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *CoRR*, abs/1702.01802.

Ekaterina Garmash and Christof Monz. 2016. Ensemble learning for multi-source neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1409–1418, Osaka, Japan. The COLING 2016 Organizing Committee.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou, and Tomás Mikolov. 2016. Fasttext.zip: Compressing text classification models. *CoRR*, abs/1612.03651.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

B. Li, Z. Wang, H. Liu, Q. Du, T. Xiao, C. Zhang, and J. Zhu. 2021. Learning light-weight translation models from deep transformer. In *National Conference on Artificial Intelligence*.

Bei Li, Ziyang Wang, Hui Liu, Yufan Jiang, Quan Du, Tong Xiao, Huizhen Wang, and Jingbo Zhu. 2020. Shallow-to-deep training for neural machine translation. *CoRR*, abs/2010.03737.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 411–416, San Diego, California. Association for Computational Linguistics.

Graham Neubig. 2011. The Kyoto free translation task. http://www.phontron.com/kftt.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

R. Pryzant, Y. Chung, D. Jurafsky, and D. Britz. 2018. JESC: Japanese-English Subtitle Corpus. *Language Resources and Evaluation Conference (LREC)*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Sandeep Subramanian, Oleksii Hrinchuk, Virginia Adams, and Oleksii Kuchaiev. 2021. NVIDIA nemo neural machine translation systems for english-german and english-russian news and biomedical tasks at WMT21. *CoRR*, abs/2111.08634.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 374–381, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Longyue Wang, Mu Li, Fangxu Liu, Shuming Shi, Zhaopeng Tu, Xing Wang, Shuangzhi Wu, Jiali Zeng, and Wen Zhang. 2021. Tencent translation system for the WMT21 news translation task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 216–224, Online. Association for Computational Linguistics.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822, Florence, Italy. Association for Computational Linguistics.

Daimeng Wei, Zongyao Li, Zhanglin Wu, Zhengzhe Yu, Xiaoyu Chen, Hengchao Shang, Jiaxin Guo, Minghan Wang, Lizhi Lei, Min Zhang, Hao Yang, and Ying Qin. 2021. HW-TSC's participation in the WMT 2021 news translation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 225–231, Online. Association for Computational Linguistics.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4207–4216, Hong Kong, China. Association for Computational Linguistics.

Xianfeng Zeng, Yijin Liu, Ernan Li, Qiu Ran, Fandong Meng, Peng Li, Jinan Xu, and Jie Zhou. 2021. WeChat neural machine translation systems for WMT21. In *Proceedings of the Sixth Conference on Machine Translation*, pages 243–254, Online. Association for Computational Linguistics.

Shuhan Zhou, Tao Zhou, Binghao Wei, Yingfeng Luo, Yongyu Mu, Zefan Zhou, Chenglong Wang, Xuanjun Zhou, Chuanhao Lv, Yi Jing, Laohu Wang, Jingnan Zhang, Canan Huang, Zhongxiang Yan, Chi Hu, Bei Li, Tong Xiao, and Jingbo Zhu. 2021. The NiuTrans machine translation systems for WMT21. In *Proceedings of the Sixth Conference on Machine Translation*, pages 265–272, Online. Association for Computational Linguistics.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, Portorož, Slovenia. European Language Resources Association (ELRA).

HW-TSC's Submissions to the WMT 2022 General Machine Translation Shared Task

Daimeng Wei, Zhiqiang Rao, Zhanglin Wu, Shaojun Li, Yuanchang Luo,
Yuhao Xie, Xiaoyu Chen, Hengchao Shang, Zongyao Li, Zhengzhe Yu,
Jinlong Yang, Miaomiao Ma, Lizhi Lei, Hao Yang, Ying Qin,
Huawei Translation Service Center, Beijing, China
{weidaimeng,raozhiqiang,wuzhanglin2,lishaojun18,luoyuanchang,
xieyuhao2,chenxiaoyu35,shanghengchao,lizongyao,yuzhengzhe,
yangjinlong7,mamiaomiao,leilizhi,yanghao30,qinying}@huawei.com

Abstract

This paper presents the submissions of Huawei Translate Services Center (HW-TSC) to the WMT 2022 General Machine Translation Shared Task. We participate in 6 language pairs, including Zh↔En, Ru↔En, Uk↔En, Hr↔En, Uk↔Cs and Liv↔En. We use Transformer architecture and obtain the best performance via multiple variants with larger parameter sizes. We perform fine-grained pre-processing and filtering on the provided large-scale bilingual and monolingual datasets. For medium and high-resource languages, we mainly use data augmentation strategies, including Back Translation, Self Training, Ensemble Knowledge Distillation, Multilingual, etc. For low-resource languages such as Liv, we use pre-trained machine translation models, and then continue training with Regularization Dropout (R-Drop). The previous mentioned data augmentation methods are also used. Our submissions obtain competitive results in the final evaluation.

1 Introduction

This paper introduces our submissions to the WMT 2022 General Machine Translation Shared Task. We participate in 6 language pairs including Chinese/English (Zh↔En), Russian/English (Ru↔En), Ukrainian/English (Uk↔En), Croatian/English (En→Hr), Ukrainian/Czech(Uk↔Cs), and Livonian/English (Liv↔En). For Zh↔En translation, we use additional in-house in-domain data, so the final submission for this language pair is unconstrained. For Liv↔En translation, although we did not use additional data, we used M2M-100 (Fan et al., 2020) as the pretrained model, and the final submission is also unconstrained. All other languages pair participate in the constrained evaluation. Our method is mainly based on previous works (Wei et al., 2020, 2021; Yang et al., 2021) but with fine-grained data cleansing techniques and language-specific optimizations.

For each language pair, we perform multi-step data cleansing on the provided dataset and only keep a high-quality subset for training. At the same time, several strategies are tested in a pipeline, including Backward (Edunov et al., 2018) and Forward (Wu et al., 2019a) Translation, Multilingual Translation (Johnson et al., 2017), Iterative Joint Training (Zhang et al., 2018), R-Drop, Pretrained NMT model, Ensemble Knowledge Distillation (Freitag et al., 2017; Li et al., 2019), Fine-Tuning (Sun et al., 2019), Ensemble (Garmash and Monz, 2016), and Post-Processing.

Our system report includes four parts. Section 2 focuses on our data processing strategies while section 3 describes our training details. Section 4 explains our experiment settings and training processes and section 5 presents the results.

2 Data

2.1 Data Source

We obtain bilingual and monolingual data from data sources such as CCMT, UN, ParaCrawl, WikiMatrix, WikiTitles, News Commentary, Leipzig Corpora, News Crawl, and Common Crawl. The amount of data we used is shown in Table 1. It should be noted that in order to obtain better performance in the general domain, we mix the monolingual data from Common Crawl and News Crawl.

2.2 Data Pre-processing

Our data processing procedure is basically the same as our method last year (Wei et al., 2021), including deduplication, XML content processing, langid (Joulin et al., 2016b,a) and fast-align (Dyer et al., 2013) filtering strategies, etc. As we use the same data pre-processing strategy as last year's, we will not go into details here.

2.3 Data Denoise

Regarding Hr↔En, the CCMatrix data is highly noisy, so more fine-grained data cleaning is nec-

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 403 - 410
December 7-8, 2022 ©2022 Association for Computational Linguistics

language pairs	Raw bi data	Filter bi data	Used mono data
Zh/En	39M	37M	En: 150M (C&N), Zh: 150M (C)
Ru/En	28M	26M	En: 160M (C&N), Ru: 160M (C&N)
Hr/En	69M	55M	Hr: 22M (N)
Uk/En	39M	36M	En: 150M (C&N), Uk: 60M (N)
Cs/Uk	8.4M	8M	Cs: 60M (C&N), Uk: 60M (N)
Liv/En	1.1k	1.1k	Liv: 50K, En: 1M

Table 1: Bilingual data sizes before and after filtering, and monolingual data used in the task. Regarding monolingual data, **N** means that the data comes from News Crawl; **C** means that the data comes from Common Crawl; and **C&N** means half of News and Common Crawl.

essary. We adopted the data denoise strategy by Wang et al. (2019, 2018). The strategy uses a small amount of high-quality data to tune the base model, and then leverages the differences between the tuned model and the baseline to score bilingual data. The score is calculated based on formula 1.

$$score = \frac{\log P(y|x;\theta_{clean}) - \log P(y|x;\theta_{noise})}{|y|} \tag{1}$$

Where θ_{noise} denotes the model trained with noisy data; θ_{clean} denotes the model after fine-tuning on a small amount of clean bilingual data, and $|y|$ denotes the length of the sentence. Higher $score$ means higher quality.

3 System Overview

Our method basically follows our previous training strategies (Wei et al., 2020, 2021), such as commonly used Back-Translation (Edunov et al., 2018), Iterative Joint Training (Zhang et al., 2018), Multilingual enhancement (Johnson et al., 2017; Kudugunta et al., 2019; Zhang et al., 2020), Data Diversification (Nguyen et al., 2020) (for details, please refer to our previous work Yang et al. (2021)), Ensemble and Fine-tuning, etc. We will not detail these strategies in this report. The following paper focuses on new strategies used in this year.

3.1 Model

We continue using Transformer (Vaswani et al., 2017) as our NMT architecture, but we do not use the four model variants as last year. For convenience, we only use a 25-6 deep model architecture. The parameters of the model are the same as Transformer-big. We just change the post-layer-normalization to the pre-layer-normalization, and increase the encoder layers to 25.

3.2 R-Drop

Dropout-like method (Srivastava et al., 2014; Gao et al., 2022) is a powerful and widely used technique for regularizing deep neural networks. Though it can help improve training effectiveness, the randomness introduced by dropouts may lead to inconsistencies between training and inference. R-Drop (Wu et al., 2021) forces the output distributions of different sub models generated by dropout be consistent with each other. Therefore, we use R-Drop training strategy to augment the baseline model for each track and reduce inconsistencies between training and inference.

3.3 Pretrained NMT Model

There are many pre-trained Sequence-to-Sequence models, such as Mbart (Liu et al., 2020), MT5 (Xue et al., 2020), M2M-100 (Fan et al., 2020), etc. These pre-trained models are very useful for ultra-low resource tasks. For the ultra-low-resource track Liv↔En, very few bilingual data (1k) is available, so we use a method similar to Adelani and Alabi (2022) to continue training on the basis of M2M-100 (418M) [1]. Since M2M-100 does not support the Liv language, we select an existing language tag (Estonian) similar to Liv to identify this language. For unknown tokens in Liv, we replace them with very low-frequent words in the vocabulary. We find this strategy effective for performance improvement.

3.4 Noised Self-Training

Self-training (Imamura and Sumita, 2018) (ST), also known as Forward translation (Wu et al., 2019b), usually refers to using a forward NMT model to translate source-side monolingual data so as to generate synthetic bilinguals, which aims at

[1] https://dl.fbaipublicfiles.com/m2m_100/418M_last_checkpoint.pt

System	WMT20	WMT21	Med20	Flores	Avg	WMT22
baseline	41.6	32.2	34.3	42.2	37.6	-
R-Drop	43.4	32.9	35.6	44.0	39.0	-
Data Rejuvenation	43.5	33.0	35.4	44.3	39.5	-
Data Diversification	44.8	33.4	35.7	44.5	39.6	-
ST+BT	45.0	33.8	36.6	45.0	40.1	46.0
Finetune & Ensemble (constrain)	-	-	-	-	-	**47.8**
Domain Data (unconstrain)	-	-	-	-	-	**49.7**

Table 2: En→Zh BLEU scores on WMT 2020 News (WMT20), WMT 2021 News (WMT21), WMT 2020 Biomedical (Med20) and Flores test sets, and their average (Avg) scores based on different training strategies. We also report part of WMT 2022 (WMT22) test set results.

System	WMT20	WMT21	Med20	Flores	Avg	WMT22
baseline	28.6	23.5	26.3	30.5	27.2	-
R-Drop	30.4	25.0	28.3	31.8	28.9	-
Data Rejuvenation	31.3	26.2	28.4	31.3	29.3	-
Data Diversification	32.5	27.8	29.5	31.9	30.4	-
ST+BT	33.3	28.1	29.6	32.0	30.7	26.0
Finetune & Ensemble (constrain)	-	-	-	-	-	**27.7**
Domain Data (unconstrain)	-	-	-	-	-	**29.8**

Table 3: Zh→En BLEU scores on WMT 2020 News (WMT20), WMT 2021 News (WMT21), WMT 2020 Biomedical (Med20) and Flores test sets, and their average (Avg) scores based on different training strategies. We also report part of WMT 2022 (WMT22) test set results.

increasing the training data size. Forward translation usually relies on beam search-based (Freitag and Al-Onaizan, 2017) decoding when generating synthetic data. He et al. (2019) find that drop-out plays an important role in ST and adding a certain noise to the original text can further improve the effect of ST, which is called Noised ST. We adopt this method during training.

3.5 Data Rejuvenation

We score all the training bilingual data through Equation 1, and filter out 10% - 20% of the data according to the score distribution. We use the remaining 80% - 90% clean data to continue training on the previous model for denoising. This strategy is particularly effective with noisy data and is used in several several languages in this task. We refer to it as Data Rejuvenation in the following.

4 Experiment Settings

We use the open-source fairseq (Ott et al., 2019) for training and sacreBLEU (Post, 2018) to measure system performances. The main parameters are as follows: Each model is trained using 8 V100 GPUs. The size of each batch is set as 2048, parameter update frequency as 4, and learning rate as 5e-4 (Vaswani et al., 2017). The number of warmup steps is 4000, and model is saved every 1000 steps. The architecture we used is described in section 3.1. We adopt dropout, and the rate varies across different language pairs. R-Drop is used in model training, and we set parameter λ to 5 for all language pairs.

5 Results and Analysis

5.1 Zh↔En

Regarding Zh↔En, we use R-Drop, Knowledge Distillation (Kim and Rush, 2016), Self Training + Back Translation, and fine-tuning. The results of Zh→En and En→Zh are shown in Tables 2 and 3.

To better measure the generalizability of our models, we also calculate BLEU on WMT Biomedical 2020 and Flores test sets (Goyal et al., 2021).

We see that R-Drop can stably bring about 1.5 BLEU improvement, and data enhancement can bring 1.0 BLEU improvement. In the final result we submitted, we only use the news test sets to fine-tune the model, but we see that it was still able to bring 1 BLEU improvement on the WMT 2022 test set.

In the end, our submission uses a combination of our domain-related in-house data and the WMT

System	WMT20	WMT21	Med20	Flores	Avg	WMT22
baseline	22.9	26.2	32.7	30.8	28.2	-
ST+BT	23.8	27.9	33.1	31.3	29.0	-
ST+BT+R2L	24.1	28.4	32.1	31.6	29.1	-
Data Rejuvenation	22.9	27.1	34.9	31.5	29.1	27.2
Common Crawl	24.1	28.6	34.5	32.7	30.0	29.4
Finetune	-	-	-	-	-	30.4
Ensemble	-	-	-	-	-	**30.8**

Table 4: En→Ru BLEU scores on WMT 2020 News (WMT20), WMT 2021 News (WMT21), WMT 2020 Biomedical (Med20) and Flores test sets, and their average (Avg) scores based on different training strategies. We also report part of WMT 2022 (WMT22) test set results.

System	WMT20	WMT21	Med20	Flores	Avg	WMT22
baseline	36.1	36.7	41.1	34.1	37.0	-
ST+BT	37.5	38.1	40.4	35.1	37.8	-
ST+BT+R2L	37.7	38.4	41.4	36.2	38.4	42.8
Data Rejuvenation	37.1	38.1	42.7	36.7	38.7	43.0
Common Crawl	37.4	38.1	42.6	36.5	38.7	43.4
Finetune	-	-	-	-	-	44.6
Ensemble	-	-	-	-	-	**45.1**

Table 5: Ru→En BLEU scores on WMT 2020 News (WMT20), WMT 2021 News (WMT21), WMT 2020 Biomedical (Med20) and Flores test sets, and their average (Avg) scores based on different training strategies. We also report part of WMT 2022 (WMT22) test set results.

data, and we find that domain-related data is critical for quality improvement. By using the extra data, we get an improvement of about 2.0 BLEU over using only the WMT data. Our final Zh→En and En→Zh submissions achieve 49.7 and 29.8 BLEU respectively.

5.2 Ru↔En

Regarding Ru↔En (Table 4 and 5), we use strategies including Iterative Self Training + Back Translation, R2L enhancement, and general domain monolingual enhancement.

We see that in addition to the average 1 BLEU improvement brought by fine-tune, the most effective strategy is adding more general domain data. On En→Ru, after the Common Crawl monolingual is added, we observe 2.0 BLEU improvement on WMT 2022 test set.

The data enhancement strategy could bring stable improvement like that in Zh↔En, with an increase of 2 BLEU compared to the baseline model in an average.

The BLEU scores of our final Ru→En and En→Ru submissions are 45.1 and 30.8 respectively.

System	En→Liv	Liv→En
M2M-100 finetune	8.0	16.0
OOV process	9.6	17.6
Multilingual	11.0	21.6
Iter Tagged BT	13.3	24.0
Noised ST	14.6	-
R-Drop	15.1	25.8
WMT22 Submission	**12.8**	**23.4**

Table 6: The results of Liv↔En for WMT 2022 dev test set. We remove overlapping sentences in the dev set that also appear in the training set.

5.3 Liv↔En

Regarding Liv↔En (Table 6), we first fine-tune the M2M-100 model with 1K bilingual data, and then replace the out-of-vocabulary (OOV) token in Liv with low-frequency sub-words in the vocabulary, we see that this strategy brings 1.6 BLEU improvement on En→Liv.

Then we use the Liv/Et and Liv/Lv data together to fine-tune the model. This strategy can bring significant improvement on both directions (1.4 BLEU on En→Liv and 4 BLEU on Liv→En. It should be pointed out that regarding En→Liv, we use additional data from Et→Liv and Lv→Liv, while for

System	dev	Flores	Avg
R-Drop	31.5	33.2	32.4
Data Rejuvenation	32.1	33.5	32.8
Sampling BT	33.2	32.9	33.1
Finetune	33.1	33.0	33.0
Ensemble	33.2	33.4	33.3
WMT22 Submission		**18.1**	

Table 7: The results of En→Hr on WMT 2022 dev test set and Flores.

Liv→En, we use data from Liv→Et and Liv→Lv to enhance the model.

We do three rounds of Tagged BT (Caswell et al., 2019) in total and observe that the improvement is still significant (an average improvement of 3 BLEU on two directions). For En→Liv, we adopt the strategy of Noised ST because we have a large amount of English monolinguals. We used 1M English monolinguals for Noised ST. We see that this strategy can bring an additional 1.3 BLEU improvement.

Additionally, we employ the R-Drop strategy during training and find that on Liv2En, this strategy brings an improvement of 1.8 BLEU.

Finally, using dev fine-tune and ensemble of 4 models, our submissions achieve 12.8 BLEU on En→Liv, and 23.4 BLEU on Liv→En.

5.4 En→Hr

The results of En→Hr are shown in Table 7. We use 22M Hr monolinguals for BT and find that the results on the dev set is different from that on the test set as the magnitude of improvements are inconsistent. The overall improvement on dev set is only 0.8 BLEU, but 3 BLEU on the test set. The main improvement is brought by data denoising. We assume that this is because the provided En2Hr bilingual data is highly noisy. Our final submission achieves 18.1 BLEU.

5.5 Uk↔En and Cs↔Uk

Regarding Uk↔En (Table 8), we conduct Sampling BT and see 2.2 BLEU improvement on Uk→En but no improvement on En↔Uk. After adding self-training data, an additional 0.5 BLEU improvement is gained on Uk→En. We then use real bilinguals data to continue training the model that have been augmented with synthetic data. This strategy further leads to an average improvement of 0.4 BLEU. We do not use dev fine-tuning but directly ensemble the 4 models. The final En→Uk and Uk→En submissions achieve 26.5 and 41.6 BLEU respectively on the WMT22 test set.

The strategy for Cs↔Uk is basically the same as that for Uk↔En, but we further apply multilingual enhancement. We use additional En→Uk data for enhancing Cs→Uk translation and En→Cs data for enhancing Uk→Cs translation. Multilingual enhancement brings 1.2 BLEU improvement on Uk→ Cs. Monolingual data augmentation also brings significant improvement. Ensemble further leads to 1 BLEU increase on Uk→Cs. Our final Cs↔Uk submissions achieve 36.0 BLEU on the WMT22 test sets.

6 Discussion

6.1 General Domain

In this year, WMT changed its focus on news domain to the broader general task, with three additional domains putting into consideration (social, conversational, and ecommerce). We also use test sets from other domains to measure the generalizability of our models.

However, for language pairs we participate in, most of the knowledge in domains other than news can only be learned from Common Crawl monolinguals. Without in-domain data, a model's performance in social, conversational and ecommerce domains can hardly be improved. We add additional bilingual data related to the three domains for the Zh↔En track and observe an average of 2.0 BLEU improvement. As a result, how to maximize the effectiveness of in-domain data is crucial.

6.2 Evaluation Method

N-gram matching metrics such as BLEU and chrF (Popović, 2015) are widely used in machine translation evaluation. However, as machine translation technology improves, relying only on BLEU to evaluate a model's performance become increasingly risky. For example, in last year's evaluation, the BLEU score of our De→En model ranks among the top, but the human evaluation results show that our model performs the worst. In this year's En→Uk evaluation, widely-used back-translation lead to no BLEU increase as shown in Table 8. So far, we are not sure whether back-translation does lead to no improvement or the improvement cannot be measured by BLEU. We believe that more researches are required on robust metrics (Sellam et al., 2020; Rei et al., 2020), reliable test set constructions, and sound human evaluation methods

System	En→Uk	Uk→En	Cs→UK	Uk→Cs
baseline	31.7	38.7	24.1	22.3
Multilingual	-	-	24.6	23.5
Sampling BT	31.7	40.9	25.7	24.2
ST + BT	31.5	41.4	25.4	23.9
Data Rejuvenation	31.9	41.8	25.7	24.2
Ensemble	32.9	41.9	26.3	25.1
WMT22 Submission	**26.5**	**41.6**	**36.0**	**36.0**

Table 8: The results of Uk↔En and Uk↔Cs for WMT 2022 dev set.

considering the great advances in NMT and subtle differences among systems.

7 Conclusion

This paper presents the submissions of HW-TSC to the WMT 2022 General Machine Translation Task. We participate in six language pairs and perform experiments with a series of pre-processing and training strategies. The effectiveness of each strategy is demonstrated. Our experiments show that in very low-resource scenarios, fine-tuning on pre-trained NMT models can significantly improve system performance. R-Drop also brings stable improvement across languages. Certainly, commonly-used data augmentation strategies are still effective for model training. Our submissions finally achieve competitive results in the evaluation.

References

David Adelani and Jesujoba et al. Alabi. 2022. A few thousand translations go a long way! leveraging pre-trained models for African news translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3053–3070, Seattle, United States. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 644–648.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 489–500.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation.

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 56–60.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *CoRR*, abs/1702.01802.

Pengzhi Gao, Zhongjun He, Hua Wu, and Haifeng Wang. 2022. Bi-simcut: A simple strategy for boosting neural machine translation.

Ekaterina Garmash and Christof Monz. 2016. Ensemble learning for multi-source neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1409–1418.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation.

Kenji Imamura and Eiichiro Sumita. 2018. Nict self-training approach to neural machine translation at nmt-2018. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 110–115.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation.

Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual NMT representations at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1565–1575, Hong Kong, China. Association for Computational Linguistics.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The niutrans machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 257–266.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation.

Xuan-Phi Nguyen, Shafiq Joty, Kui Wu, and Ai Ti Aw. 2020. Data diversification: A simple strategy for neural machine translation. In *Advances in Neural Information Processing Systems*, volume 33, pages 10018–10029. Curran Associates, Inc.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 374–381.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wei Wang, Isaac Caswell, and Ciprian Chelba. 2019. Dynamically composing domain-data selection with clean-data selection by "co-curricular learning" for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1282–1292, Florence, Italy. Association for Computational Linguistics.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018. Denoising neural machine translation training with trusted data and online data selection. pages 133–143.

Daimeng Wei, Zongyao Li, Zhanglin Wu, Zhengzhe Yu, Xiaoyu Chen, Hengchao Shang, Jiaxin Guo, Minghan Wang, Lizhi Lei, Min Zhang, Hao Yang, and Ying Qin. 2021. HW-TSC's participation in the WMT 2021 news translation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 225–231, Online. Association for Computational Linguistics.

Daimeng Wei, Hengchao Shang, Zhanglin Wu, Zhengzhe Yu, Liangyou Li, Jiaxin Guo, Minghan Wang, Hao Yang, Lizhi Lei, Ying Qin, and Shiliang Sun. 2020. HW-TSC's participation in the WMT 2020 news translation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 293–299, Online. Association for Computational Linguistics.

Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. 2021. R-drop: Regularized dropout for neural networks. *Advances in Neural Information Processing Systems*, 34:10890–10905.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019a. Exploiting monolingual data at scale for neural machine translation.

In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4205–4215.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019b. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4207–4216.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer.

Hao Yang, Zhanglin Wu, Zhengzhe Yu, Xiaoyu Chen, Daimeng Wei, Zongyao Li, Hengchao Shang, Minghan Wang, Jiaxin Guo, Lizhi Lei, Chuanfei Xu, Min Zhang, and Ying Qin. 2021. HW-TSC's submissions to the WMT21 biomedical translation task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 879–884, Online. Association for Computational Linguistics.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Vega-MT: The JD Explore Academy Translation System for WMT22

Changtong Zan[ℜ,♭]🧸, Keqin Peng[♯]🧸, Liang Ding[ℜ]🧸, Baopu Qiu[♮], Boan Liu[♢], Shwai He[△]
Qingyu Lu[♡], Zheng Zhang[♢], Chuang Liu[♢], Weifeng Liu[♭], Yibing Zhan[ℜ], Dacheng Tao[ℜ]
[ℜ]JD Explore Academy, JD.com Inc.
[♭]China University of Petroleum (East China) [♯]Beihang University [♮]Nanjing University
[♢]Wuhan University [△]University of Electronic Science and Technology of China [♡]Southeast University
📧 dingliang1@jd.com

Abstract

We describe the JD Explore Academy's submission of the WMT 2022 shared task on general machine translation. We participated in all high-resource tracks and one medium-resource track, including Chinese↔English (Zh↔En), German↔English (De↔En), Czech↔English (Cs↔En), Russian↔English (Ru↔En), and Japanese↔English (Ja↔En). **[Method]** We push the limit of our previous work – bidirectional training (Ding et al., 2021d) for translation by scaling up two main factors, *i.e.* language pairs and model sizes, namely the **Vega-MT** system. As for language pairs, we scale the "bidirectional" up to the "multidirectional" settings, covering all participating languages, to exploit the common knowledge across languages, and transfer them to the downstream bilingual tasks. As for model sizes, we scale the Transformer-BIG up to the extremely large model that owns nearly 4.7 Billion parameters, to fully enhance the model capacity for our Vega-MT. Also, we adopt the data augmentation strategies, *e.g.* cycle translation (Ding and Tao, 2019) for monolingual data, and bidirectional self-training (Ding and Tao, 2021) for bilingual and monolingual data, to comprehensively exploit the bilingual and monolingual data. To adapt our Vega-MT to the general domain test set, generalization tuning is designed. **[Results]** Based on the official automatic scores[*] of constrained systems, in terms of the SACREBLEU (Post, 2018) shown in Figure 1, we got the 1st place in {Zh-En (33.5), En-Zh (49.7), De-En (33.7), En-De (37.8), Cs-En (54.9), En-Cs (41.4) and En-Ru (32.7)}, 2nd place in {Ru-En (45.1) and Ja-En (25.6)}, and 3rd place in {En-Ja(41.5)}, respectively; W.R.T the **COMET** (Rei et al., 2020), we got the 1st place in {Zh-En (45.1), En-Zh (61.7), De-En (58.0), En-De (63.2), Cs-En (74.7), Ru-En (64.9), En-Ru (69.6) and En-Ja (65.1)}, 2nd place in {En-Cs (95.3) and Ja-En (40.6)}, respectively. Models will be released to facilitate the MT community through GitHub[†] and OmniForce Platform[‡].

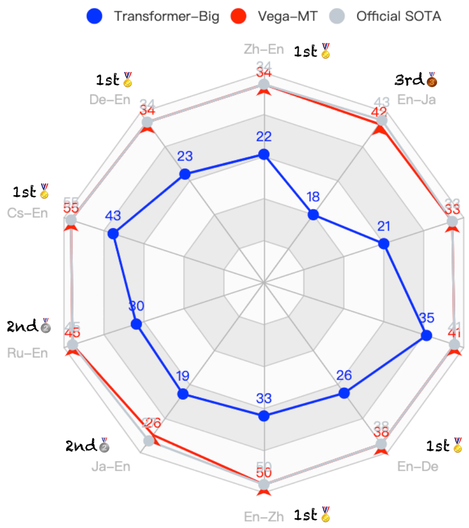


Figure 1: Vega-MT achieves 7 state-of-the-art BLEU points out of 10 high-resource translation tasks among all constrained systems, and significantly outperforms the competitive Transformer-BIG baselines.

1 Introduction

In this year's WMT general translation task, our Vega-MT translation team participated in 10 shared tasks, including Chinese↔English (Zh↔En), German↔English (De↔En), Czech↔English (Cs↔En), Russian↔English (Ru↔En), and Japanese↔English (Ja↔En). We use the same model architectures, data strategies and corresponding techniques for all tasks.

🧸 Equal contribution. Work was done when Changtong and Keqin were interning at JD Explore Academy.

[*] https://github.com/wmt-conference/wmt22-news-systems/tree/main/scores

[†] https://github.com/JDEA-NLP/Vega-MT

[‡] OmniForce Platform will be launched by JD Explore Academy

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 411 - 422
December 7-8, 2022 ©2022 Association for Computational Linguistics

We aim to leverage the cross-lingual knowledge through pretraining (PT) to improve the high-resource downstream bilingual tasks. Although recent works (Song et al., 2019; Lewis et al., 2020; Liu et al., 2020b; Wang et al., 2022) attempt to leverage sequence-to-sequence PT for neural machine translation (NMT; Bahdanau et al., 2015a; Gehring et al., 2017; Vaswani et al., 2017a) by using a large amount of unlabeled (*i.e.* monolingual) data, Zan et al. (2022b) show that it usually fails to achieve notable gains (sometimes, even worse) on resource-rich NMT on par with their random-initialization counterpart, which is consistent with our preliminary experiments. Ding et al. (2021d) show that bidirectional pretrained model as initialization for downstream bilingual tasks could consistently achieve significantly better performance. It is natural to assume that scaling the "bidirectional" to the "multidirectional" setting with {1) *multilingual pretraining* and 2) *large enough model capacity*} could benefit the downstream resource-rich bilingual translations. Tran et al. (2021) and Lin et al. (2020) also provide empirical evidences to support our motivation of supervised multilingual pretraining. Different from Tran et al. (2021) that explores the effectiveness of multilingual training, we show that further tuning on the bilingual downstream task provide more in-domain knowledge and thus could gain better translation quality. Compared with Lin et al. (2020), our model do not require any alignment information during pretraining, which will consume more extra time and computation resources, making our strategy flexible to be applied to any language.

For model frameworks in §2.1, we tried autoregressive neural machine translation, including Transformer-BIG and -XL (Vaswani et al., 2017b), and non-autoregressive translation models (Gu et al., 2018), where the Transformer-XL is employed as the foundation model and autoregressive BIG and non-autoregressive models are used during augmenting. For the core training strategy of our Vega-MT, we cast the multilingual pretraining as foundation models in §2.2, including MULTI-DIRECTIONAL PRETRAINING (§2.2.1) and SPECIFIC-DIRECTIONAL FINETUNING (§2.2.2). For data augmentation strategies, we employ CYCLE TRANSLATION (§2.3.1) and BIDIRECTIONAL SELF-TRAINING (§2.3.2) for both monolingual and parallel data. In order to adapt our Vega-MT to the general domains, we employ GREEDY BASED ENSEMBLING (§2.4.1), GENERALIZATION FINETUNING (§2.4.2) and POST-PROCESSING (§2.4.3) strategies.

	$\mathcal{M}_{\textbf{Base}}$	$\mathcal{M}_{\textbf{Big}}$	$\mathcal{M}_{\textbf{XL}}$
#Stack	6	6	24
#Hidden_Size	512	1024	2048
#FFN_Size	2048	4096	16384
#Heads	8	16	32

Table 1: Model differences among base ($\mathcal{M}_{\textbf{Base}}$), big ($\mathcal{M}_{\textbf{Big}}$) and extremely large ($\mathcal{M}_{\textbf{XL}}$).

The subsequent paper is designed as follows. We introduce the major approaches we used in Section 2. In Section 3, we provide the data description. We also present the experimental settings and results in Section 4. Conclusions are described in Section 5.

2 Approaches

2.1 Neural Machine Translation Frameworks

The neural machine translation task aims to transform a source language sentence into the target language with a neural network. There are several generation paradigms for translation, *e.g.* Autoregressive Translation (AT, Bahdanau et al., 2015b; Vaswani et al., 2017b) and Non-Autoregressive Translation (NAT, Gu et al., 2018).

Autoregressive Translation Given a source sentence $\mathbf{x}$, an NMT model generates each target word $\mathbf{y}_t$ conditioned on previously generated ones $\mathbf{y}_{<t}$. Accordingly, the probability of generating $\mathbf{y}$ is computed as:

$$p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}, \mathbf{y}_{<t}; \theta) \tag{1}$$

where T is the length of the target sequence and the parameters θ are trained to maximize the likelihood of a set of training examples according to $\mathcal{L}(\theta) = \arg\max_{\theta} \log p(\mathbf{y}|\mathbf{x}; \theta)$. Typically, we choose Transformer (Vaswani et al., 2017b) as its state-of-the-art performance and scalability. We carefully employ the standard Transformer-BASE ($\mathcal{M}_{\textbf{Base}}$) and Transformer-BIG ($\mathcal{M}_{\textbf{Big}}$) in the preliminary studies, and also scale the framework up to an extremely large setting (Tran et al., 2021) – Transformer-XL ($\mathcal{M}_{\textbf{XL}}$) to maintain powerful

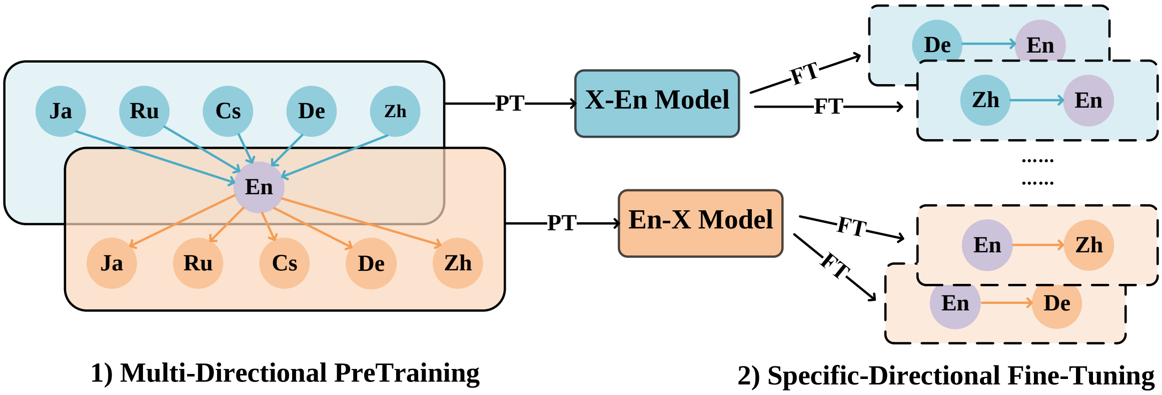


Figure 2: The schematic structure of the two main stages of the Vega-MT.

model capacity (see Table 1) . In Vega-MT, we utilized the autoregressive translation (AT) model with $\mathcal{M}_{\textbf{Big}}$ and $\mathcal{M}_{\textbf{XL}}$ for multi-directional pretraining (§2.2.1), specific-directional finetuning (§2.2.2), bidirectional self-training (§2.3.2) and generalization fine-tuning (§2.4.2) as its powerful modelling ability and generation accuracy.

Non-Autoregressive Translation Different to autoregressive translation (Bahdanau et al., 2015b; Vaswani et al., 2017b, AT) models that generate each target word conditioned on previously generated ones, non-autoregressive translation (Gu et al., 2018, NAT) models break the autoregressive factorization and produce the target words in parallel. Given a source sentence $\mathbf{x}$, the probability of generating its target sentence $\mathbf{y}$ with length T is defined by NAT as:

$$p(\mathbf{y}|\mathbf{x}) = p_L(T|\mathbf{x};\theta)\prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x};\theta) \qquad (2)$$

where $p_L(\cdot)$ is a separate conditional distribution to predict the length of target sequence. Typicallly, most NAT models are implemented upon the framework of $\mathcal{M}_{\textbf{Base}}$. We utilized the NAT for bidirectional self-training (§2.3.2) as NAT can nicely avoid the error accumulation problems during generation, and generate diverse synthetic samples. Also, we employ several advanced structure (Gu et al., 2019; Ding et al., 2020) (*Levenshtein* with source local context modelling) and advanced training strategies (Ding et al., 2021a,b,c, 2022b; Ding, 2022) to obtain high quality and diverse translations.

2.2 Multidirectional Pretraining as Foundation Models

This section illustrates how we scale the "bidirectional" training in Ding et al. (2021d) up to "multi-directional" pretraining with all high-resource parallel corpora, including Zh, De, Cs, Ru, Ja to/from En. The pretrained foundation models will be fine-tuned for the downstream specific-directional task, *e.g.* Zh-En. Such two-stage scheme is shown in Figure 2.

2.2.1 Multi-Directional Pretraining

Recent works on real-world `WMT` translation datasets have verified that it is possible to transfer the pretrained cross-lingual knowledge to the downstream tasks with the pretrain-finetune paradigm, hence improving performance and generalization ability (Ding et al., 2022b,a; Wang et al., 2020a).

Here, we propose multi-directional pretraining by extending Bidirectional Pretraining (Ding et al., 2021d, BiT) to utilize multiple translation corpora of different languages. Compared with BiT, multi-directional pretraining could utilize the cross-lingual knowledge among more languages, thus further exploiting the cross-language knowledge and facilitating the downstream transferring. The main modifications could be summarized twofold:

1) We increase language numbers to utilize the cross-lingual knowledge of various languages. The straight setting for multi-directional pretraining is multi-lingual translation, which is divided into Many-to-Many (M2M), One-to-Many (O2M), and Many-to-One (M2O), according to the language number that the model supports. M2M has potential of capturing more cross-

lingual knowledge from $N * N$ pairs compared with $N * 1/1 * N$ pairs of M2O/O2M but usually leads to worse performance because of the imbalanced language data distribution question (Freitag and Firat, 2020). Inspired by (Tran et al., 2021), we focus on pretraining two separate systems, including English-to-Many and Many-to-English. We also prepend the corresponding language token to source & target sentences.

2) We further expand model size to an extremely large setting. While enjoying the benefit of cross-lingual knowledge transferring, the difficulty of modeling extremely large-scale data and language-specific feature pushes us to enlarge Transformer-BIG to an extremely large size (4.7 Billion parameters, see Table 1). This ensures our models are capable of better mastering multiple translation corpus.

2.2.2 Specific-Directional Finetuning

The off-target problem, which widely exits in multilingual translation systems (Yang et al., 2021), indicates model often generates the translation with some non-target words. To reduce non-target word translation ratio in multi-directional pretrained models, we consider a two-stage specific-directional finetuning strategy. As shown in Figure 2, the English source/target model is tuned with an English source/target bilingual corpus.

Specifically, we first replace the multilingual embedding with a bilingual one. To fit model and bilingual vocabulary, we freeze all parameters of the Transformer backbone and only tune embedding layers in this stage. Next, we employ full model finetuning on large-scale translation corpus. This allows the model to fully adapt to the specific directional translation task, thus further achieving gains. To balance both finetune stages, we set the ratios of update step as $1 : 4$ for embedding- and full model-tuning, respectively.

For future work during specific directional finetuning, it will be interesting to design tuning data order (Liu et al., 2020a; Zhou et al., 2021) by leveraging the learning difficulty of each training sample estimated in the pretraining stage.

2.3 Data Augmentation Strategies

In Vega-MT, we consider augmenting both the parallel and monolingual data comprehensively. Specifically, we employ the cycle translation (Ding and Tao, 2019) for regenerating the low-quality *monolingual data*, and adopt bidirectional self-training (Ding and Tao, 2021) to distill, diversify *both the monolingual and parallel data*.

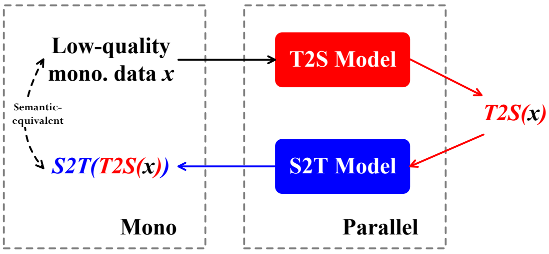


Figure 3: The Cycle Translation process, into which we feed the low quality monolingual data x, and then correspondingly obtain the improved data $\mathcal{CT}(x)$ (denoted as $S2T(T2S(x))$). Note that models marked in red and blue represent the target-to-source and source-to-target model trained with $\mathcal{M}_{\textbf{Big}}$. The dotted double-headed arrow between the input **x** and the final output $\mathcal{CT}(x)$ means they share the semantic but differ in fluency.

#	Cycle Translated Sentence "1"→"2"
1	*She stuck to her principles even when some suggest that in an environment often considered devoid of such thing there are little point.*
2	*She insists on her own principles, even if some people think that it doesn't make sense in an environment that is often considered to be absent.*

Table 2: Example of difference between original sentence (line 1) and cycle translated result (line 2). Pretrained BERT model using all available English corpora show that the $\mathcal{L}oss$ decreased from 6.98 to 1.52.

2.3.1 Cycle Translation for Mono. Data

There is a large amount of monolingual data incomplete or grammatically incorrect. To fully leverage such part of monolingual data for better data augmentation, *e.g.* back translation (Sennrich et al., 2016) or sequence -level knowledge distillation (Kim and Rush, 2016), we adopt Cycle Translation (Ding and Tao, 2019) (denoted as $\mathcal{CT}(\cdot)$, as Figure 3) to improve the monolingual data below the quality-threshold (the latter 50% will be cycle translated according to Ding and Tao (2019)'s optimal setting). We give an example in Table 2 to clearly show how the cycle translation improves the quality of the sentence.

2.3.2 Bidirectional Self-Training for Both Mono&Para Data

Currently, data-level methods have attracted the attention of the community, including exploiting the parallel and monolingual data. The most representative approaches include:

- Back Translation (**BT**, Sennrich et al. 2016) introduces the target-side monolingual data by translating with an inverse translation model, and combines the synthetic data with parallel data;

- Knowledge Distillation (**KD**, Kim and Rush 2016) generates the synthetic data with sequence-level knowledge distillation;

- Data Diversification (**DD**, Nguyen et al. 2020) diversifies the data by applying KD and BT on parallel data.

Clearly, self-training is at the core of above approaches, that is, they generate the synthetic data either from source to target or reversely, with either monolingual or bilingual data.

To this end, we employ the bidirectional self-training (Ding and Tao, 2021; Liao et al., 2020) strategy for both parallel and monolingual data (including source and target, respectively). Specifically, baseline AT models with $\mathcal{M}_{\textbf{Big}}$ setting and NAT models with $\mathcal{M}_{\textbf{Base}}$ setting are trained with original (distilled for NAT) parallel data in the first iteration, and based on these forward- and backward-teachers, all available source & target language sentences can be used to generate the corresponding synthetic target & source sentences. The authentic and synthetic data (generated by AT and NAT models) are then concatenated to train the second round AT and NAT models. We run the bidirectional self-training by totally 2 rounds for each translation direction. And for each round, we train 3 forward- and 3 backward- AT models, and 1 forward- and backward- NAT models to perform self-training. In this way, the amount of bidirectional synthetic data will be 8x larger than the original parallel and monolingual data.

2.4 Generalization Adaptation for Downstream Translation

To adapt Vega-MT to the general domain translation task, we employ several strategies, including ensembling, generalization finetuning, and post-processing. Note that in our preliminary study, we find that noisy channel reranking with the target-to-source MT model and language model does not work in our setting, thus we have not reranked the results in the final submission.

Algorithm 1: Generalization Finetuning with Iteratively Transductive Ensemble

```
Input: Single Model $M_n$,
       General Seed $D$=$\{D_1, D_2..D_k\}$,
       Ensemble $N$ models $E_N$.
Output: New Model $M_n^{'}$
$t := 0$
while not convergence do
    Translate $D_1$ with $E_N$ and get $D_1^{E_N}$
    ..
    Translate $D_k$ with $E_N$ and get $D_k^{E_N}$
    $D^{E_N} = D_1^{E_N} \cup ..D_k^{E_N}$
    Train $M_n$ on $D \cup D^{E_N}$ and get $M_n^{'}$,
      then $M_n = M_n^{'}$
    $t := t + 1$
end
```

SRC	*Siltalan edellinen kausi liigassa oli 2006-07*
HYP	*Siltala's previous season in the league was 2006 at 07*
+post	*Siltala's previous season in the league was 2006-07*

Table 3: Example of the effectiveness of post-processing in handling inconsistent number translation.

2.4.1 Greedy Based Ensembling

Greedy based ensembling adopts an easy operable greedy-base strategy to search for a better single model combinations on the development set, which consistently shows better performance than simply average in our preliminary study, therefore we technically follow the instruction of Deng et al. (2018) to choose the optimal combination of checkpoints to enhance the generalization and boost performance of the final model. We refer to this method as "Ensemble" in the following.

2.4.2 Generalization Finetuning

As the general domain evaluation is on multi-domain directions, *i.e.* containing (up to) four dif-

Languages	**# Sents**	**# Ave. Len.**
	Parallel	
Zh-En	46,590,547	22.8/27.1
De-En	292,020,383	22.9/21.7
Cs-En	88,244,832	20.5/19.9
Ru-En	98,454,430	28.5/27.8
Ja-En	28,943,024	26.2/28.0
	Monolingual	
En	1,384,791,758	21.3
Zh	1,346,538,572	25.8
De	5,612,161,001	23.2
Cs	444,049,843	19.7
Ru	8,351,860,471	28.5
Ja	5,534,872,418	27.9

Table 4: Data statistics after pre-processing.

ferent domains, we design generalization finetuning strategy to transductively finetune (Wang et al., 2020b) on each domain, and ensemble them into one single model, to empower the general translation ability. The proposed generalization finetuning is shown in Algorithm 1. The main difference from Multi-Model & Multi-Iteration Transductive Ensemble (Wang et al., 2021) is that the k_{th} domain seed D_k is extracted from the test set using heuristic artificial knowledge.

2.4.3 Post-Processing

In addition to general post-processing strategies (*e.g.* de-BPE), we also employ a post-processing algorithm (Wang et al., 2018) for inconsistent number, date translation, for example, "*2006-07*" might be translated to the wrong translation "*2006 at 07*". Our post-processing algorithm will search for the best matching number string from the source sentence to replace these types of errors (see Table 3). Besides, we also conduct punctuation conversion, including convert quotation marks to German double-quote style (Czech, German), convert punctuation to language-specific characters (Japanese, Chinese).

3 Data Preparation

We participated in translation of all high-resource tracks and one medium-resource track, including Chinese↔English (Zh↔En), German↔English (De↔En), Czech↔English (Cs↔En), Russian↔English (Ru↔En), and Japanese↔English (Ja↔En).

In this section, we take the En↔Zh translation as example and describe how to prepare the training data. The setting is the same for other language pairs. We use all available parallel corpus for En↔Zh [§], including ParaCrawl v9, News Commentary v16, Wiki Titles v3, UN Parallel Corpus V1.0, CCMT Corpus, WikiMatrix and Back-translated news. For monolingual data, we randomly sample from "News Crawl" and "Common Crawl". The final corpus statistics are presented in Table 4.

To improve the quality of parallel data, we further propose to filter the low-quality samples. First, we remove the sentence pair which is predicted as wrong language with `Fasttext` (Joulin et al., 2017, 2016). Second, we replace unicode punctuation and also normalize punctuation with `mosesdecoder`. We also remove duplicate sentence pairs and filter out sentences with illegal characters. For length, we remove sentences longer than 250 words and with a source/target length ratio exceeding 3.

4 Experiments

Settings We use the extremely large Transformer ($\mathcal{M}_{\textbf{XL}}$) for all tasks and Transformer-BIG ($\mathcal{M}_{\textbf{BIG}}$) for bilingual baselines. For $\mathcal{M}_{\textbf{BIG}}$, we empirically adopt large batch strategy (Edunov et al., 2018) (*i.e.* 458K tokens/batch) to optimize the performance. The learning rate warms up to 1×10^{-7} for 10K steps, and then decays for 70K steps with the cosine schedule. For regularization, we tune the dropout rate from [0.1, 0.2, 0.3] based on validation performance, and apply weight decay with 0.01 and label smoothing with $\epsilon = 0.1$. We use Adam optimizer (Kingma and Ba, 2015) to train models. We evaluate the performance on an ensemble of last 10 checkpoints to avoid stochasticity. For the main model $\mathcal{M}_{\textbf{XL}}$, we adopt 1M Tokens/Batch to optimize the performance both in multilingual pretraining and bilingual finetuning. We set 0.1 as the label smoothing ratio, 4000 as warm-up steps, and 1e-3 as the learning rate. We optimize Vega-MT with Adam (Kingma and Ba, 2015). We use 100k updates for multi-directional pretraining, 40k updates for each specific-directional finetuning. For

[§] both parallel and monolingual corpus can be obtained from`https://www.statmt.org/wmt22/translation-task.html`

	Zh-En			En-Zh		
Models	**W21 test**	**W22 test**	**Δ**	**W21 test**	**W22 test**	**Δ**
Transformer-BIG w/ Para.	25.3	21.9	-	25.9	33.2	-
Multi-Directional PT	28.4	25.1	*+3.2*	27.1	35.7	*+1.9*
+Specific-Directional FT	29.5	26.7	*+4.3*	27.4	36.6	*+3.6*
+Bidirect. Self-Training	30.8	29.0	*+6.3*	29.7	40.7	*+5.7*
+Ensemble	**31.1**	29.8	*+6.7*	30.4	41.3	*+6.4*
+Generalization FT	30.3	**33.5**	*+8.3*	30.6	44.1	*+9.0*
+Post-Processing	30.5	**33.5**	***+8.4***	**33.6**	**49.7**	***+13.3***

Table 5: **Ablation studies of each component on Zh↔En** general translation task in terms of SacreBLEU.We select Transformer-BIG only trained with official parallel data as the baseline.

Models	**Zh→En**	**De→En**	**Cs→En**	**Ru→En**	**Ja→En**	**Δ**
Baseline	21.9	23.0	42.5	30.2	19.0	-
Vega-MT	**33.5**	**33.7**	**54.9**	45.1	25.6	***+11.2***
Best Official	33.5	33.7	54.9	**45.1**	**26.6**	
Models	**En→Zh**	**En→De**	**En→Cs**	**En→Ru**	**En→Ja**	**Δ**
Baseline	33.2	26.4	34.8	20.8	17.9	-
Vega-MT	**49.7**	**37.8**	**41.4**	**32.7**	41.5	***+14.0***
Best Official	49.7	37.8	41.4	32.7	**42.5**	

Table 6: **SacreBLEU-Scores of our submissions in WMT2022 general translation task.** “Baseline” indicates the performance of the baseline systems. And “Best Official” denotes the best results of constrained systems in each direction.

evaluation, we select SacreBLEU (Post, 2018) as the metric for all tasks. `news-test2020` and `news-test2021` are selected for validation and test respectively.

All parallel data will be used in the multi-directional PT stage, and during specific-directional FT, corresponding bilingual data augmented by bidirectional self-training are utilized. Each sentence are jointly tokenized in to sub-word units with SentencePiece (Kudo and Richardson, 2018), which is trained on all concatenated multilingual parallel data for Transformer-XL with merge operation 120K at the pretraining stage, and during finetuning stage, is trained on corresponding bilingual data with merge operation 60K for English and 75K for other languages. And for each baseline with Transformer-BIG, the joint bilingual vocab size is 80K. During pretraining, we select the sample with temperature-based method (T=5) to preserve the representation of relatively low-resource language, *e.g.* Japanese. We grid-search the beam size within the range of [3,4,5,..,8] on validation set for each translation task. All models are trained on 32 DGX-SuperPOD A100 GPUs for about two weeks pre-training and five days fine-tuning.

Main Results To illustrate the effectiveness of each strategy in our Vega-MT, we report the ablation results in Table 5 on Zh↔En tasks. Clearly, directly generating the translations with the multi-directional pretrained model could obtain average +3.2 and +1.9 BLEU improvements for Zh-En and En-Zh, respectively, which is consistent with the findings of Tran et al. (2021). We show that tuning on downstream bilingual data could further improve the translation by +1.4 BLEU points, showing the necessity of bridging the cross-lingual gap with in-domain learning during leveraging multilingual pretrain (Zan et al., 2022a). Bidirectional self-training actually contains several strategies, *e.g.* back translation, distillation and data diversification, and we empirically show that such data augmentation strategy nicely complement pretraining, which is also verified by Liu et al. (2021). Other strategies could consistently enhance the translation performance besides the generalization FT for the news domain

Models	Zh→En	De→En	Cs→En	Ru→En	Ja→En	Δ
Baseline	16.5	3.5	40.1	8.5	21.5	-
Vega-MT	**45.1**	**58.0**	**74.7**	**64.9**	40.6	***+38.6***
Best Official	45.1	58.0	74.7	64.9	**42.0**	
Models	**En→Zh**	**En→De**	**En→Cs**	**En→Ru**	**En→Ja**	**Δ**
Baseline	26.6	-40.6	66.9	-1.4	42.1	-
Vega-MT	**61.7**	**63.2**	95.3	**69.6**	**65.1**	***+52.3***
Best Official	61.7	63.2	**96.0**	69.6	65.1	

Table 7: **COMET-Scores of our submissions in WMT2022 general translation task.** "Baseline" indicates the performance of the baseline systems. And "Best Official" denotes the best results of constrained systems in each direction.

test2021, where the Zh-En model decreases the BLEU scores (-0.8 BLEU) because the generalization FT is designed and tuned for the general domain test2022.

Table 6 and Table 7 show the final submissions in terms of SacreBLEU and COMET scores, including Zh, De, Cs, Ru and Ja to/from En, listing the baseline and our final submissions. We also report the best official scores among all constrained systems "Best Official" as reference. As seen, SacreBLEU and COMET results show identical trends, where our Vega-MT outperforms baseline Transformer-BIG by +11.2/ +38.6 and +14.0/ +52.3 BLEU/ COMET points, showing the effectiveness and universality of our model. Interestingly, we observe that the improvements upon En-X are more significant than that of X-En, which will be investigated in our future work. For more system rankings, please refer Table 8 and Table 9 in Appendix for SacreBLEU and COMET results, respectively.

5 Conclusion

This paper presents the JD Explore Academy machine translation system Vega-MT for WMT 2022 shared tasks on general machine translation. We investigate various frameworks, including autoregressive and non-autoregressive Transformer with BASE, BIG and XL settings, respectively, to build strong baseline models. Then we push the limit of bidirectional training by scaling up two main factors, *i.e.* language pairs and model scales, to develop the powerful foundation Vega-MT model. Also, the popular data augmentation methods, *e.g.* cycle translation and bidirectional self-training, are combined to improve their performance. We carefully design the generalization adaptation strategies to further improve the multi-domain performance. Among all participated constrained systems, our Vega-MT won 7 champions, 2 runners-up and 1 third place w.r.t sacreBLEU. And according to the COMET, we won 8 champions and 2 runners-up.

Acknowledgments

The authors wish to thank the organizers of WMT2022 for their great efforts in the organization, and their prompt responses to our questions. The authors are grateful to the anonymous reviewers for their insightful comments and careful proofreading. The authors also specially thank Yukang Zhang (JD Explore Academy), who kindly supports us by maintaining a stable computing platform.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015a. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015b. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Yongchao Deng, Shanbo Cheng, Jun Lu, Kai Song, Jingang Wang, Shenglan Wu, Liang Yao, Guchun Zhang, Haibo Zhang, Pei Zhang, Changfeng Zhu, and Boxing Chen. 2018. Alibaba's neural machine translation systems for WMT18. In *WMT*.

Liang Ding. 2022. *Neural Machine Translation with Fully Information Transformation*. Ph.D. thesis, The University of Sydney.

Liang Ding, Keqin Peng, and Dacheng Tao. 2022a. Improving neural machine translation by denoising training. *arXiv preprint*.

Liang Ding and Dacheng Tao. 2019. The University of Sydney's machine translation system for WMT19. In *WMT*.

Liang Ding and Dacheng Tao. 2021. The USYD-JD speech translation system for IWSLT2021. In *IWSLT*.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao, and Zhaopeng Tu. 2021a. Progressive multi-granularity training for non-autoregressive translation. In *findings of ACL*.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao, and Zhaopeng Tu. 2021b. Rejuvenating low-frequency words: Making the most of parallel data in non-autoregressive translation. In *ACL*.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao, and Zhaopeng Tu. 2021c. Understanding and improving lexical choice in non-autoregressive translation. In *ICLR*.

Liang Ding, Longyue Wang, Shuming Shi, Dacheng Tao, and Zhaopeng Tu. 2022b. Redistributing low-frequency words: Making the most of monolingual data in non-autoregressive translation. In *ACL*.

Liang Ding, Longyue Wang, Di Wu, Dacheng Tao, and Zhaopeng Tu. 2020. Context-aware cross-attention for non-autoregressive translation. In *COLING*.

Liang Ding, Di Wu, and Dacheng Tao. 2021d. Improving neural machine translation by bidirectional training. In *EMNLP*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *EMNLP*.

Markus Freitag and Orhan Firat. 2020. Complete multilingual neural machine translation. In *WMT*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML*.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *ICLR*.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *NeurIPS*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *EACL*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *EMNLP*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.

Baohao Liao, Yingbo Gao, and Hermann Ney. 2020. Multi-agent mutual learning at sentence-level and token-level for neural machine translation. In *Findings of EMNLP 2020*.

Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. 2020. Pre-training multilingual neural machine translation by leveraging alignment information. In *EMNLP*.

Xuebo Liu, Houtim Lai, Derek F. Wong, and Lidia S. Chao. 2020a. Norm-based curriculum learning for neural machine translation. In *ACL*.

Xuebo Liu, Longyue Wang, Derek F. Wong, Liang Ding, Lidia S. Chao, Shuming Shi, and Zhaopeng Tu. 2021. On the complementarity between pre-training and back-translation for neural machine translation. In *EMNLP*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020b. Multilingual denoising pre-training for neural machine translation. *TACL*.

Xuan-Phi Nguyen, Shafiq Joty, Kui Wu, and Ai Ti Aw. 2020. In *NeurIPS*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *WMT*.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *EMNLP*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *ACL*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *ICML*.

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. Facebook AI's WMT21 news translation task submission. In *WMT*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017a. Attention is all you need. In *NeurIPS*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017b. Attention is all you need. In *NeurIPS*.

Longyue Wang, Mu Li, Fangxu Liu, Shuming Shi, Zhaopeng Tu, Xing Wang, Shuangzhi Wu, Jiali Zeng, and Wen Zhang. 2021. Tencent translation system for the WMT21 news translation task. In *WMT*.

Longyue Wang, Zhaopeng Tu, Xing Wang, Li Ding, Liang Ding, and Shuming Shi. 2020a. Tencent AI lab machine translation systems for WMT20 chat translation task. In *WMT*.

Qiang Wang, Bei Li, Jiqiang Liu, Bojian Jiang, Zheyang Zhang, Yinqiao Li, Ye Lin, Tong Xiao, and Jingbo Zhu. 2018. The NiuTrans machine translation system for WMT18. In *WMT*.

Wenxuan Wang, Wenxiang Jiao, Yongchang Hao, Xing Wang, Shuming Shi, Zhaopeng Tu, and Michael Lyu. 2022. Understanding and improving sequence-to-sequence pretraining for neural machine translation. In *ACL*.

Yiren Wang, Lijun Wu, Yingce Xia, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2020b. Transductive ensemble learning for neural machine translation. In *AAAI*.

Yilin Yang, Akiko Eriguchi, Alexandre Muzio, Prasad Tadepalli, Stefan Lee, and Hany Hassan. 2021. Improving multilingual translation by representation and gradient regularization. In *EMNLP*.

Changtong Zan, Liang Ding, Li Shen, Yu Cao, Weifeng Liu, and Dacheng Tao. 2022a. Bridging cross-lingual gaps during leveraging the multilingual sequence-to-sequence pretraining for text generation. *arXiv preprint*.

Changtong Zan, Liang Ding, Li Shen, Yu Cao, Weifeng Liu, and Dacheng Tao. 2022b. On the complementarity between pre-training and random-initialization for resource-rich machine translation. In *COLING*.

Lei Zhou, Liang Ding, Kevin Duh, Shinji Watanabe, Ryohei Sasano, and Koichi Takeda. 2021. Self-guided curriculum learning for neural machine translation. In *IWSLT*.

pair	system	id	is_constrained	metric	score
En-Cs	Lan-Bridge	551	FALSE	bleu-B	45.6
En-Cs	JDExploreAcademy	829	TRUE	bleu-B	**41.4**
En-Cs	CUNI-DocTransformer	800	TRUE	bleu-B	39.8
En-Cs	CUNI-Bergamot	734	TRUE	bleu-B	38.6
En-Cs	CUNI-Transformer	761	TRUE	bleu-B	37.7
pair	**system**	**id**	**is_constrained**	**metric**	**score**
En-De	JDExploreAcademy	843	TRUE	bleu-A	**37.8**
En-De	Lan-Bridge	549	FALSE	bleu-A	36.1
En-De	PROMT	694	FALSE	bleu-A	36.1
En-De	OpenNMT	207	FALSE	bleu-A	35.7
pair	**system**	**id**	**is_constrained**	**metric**	**score**
En-Ja	NT5	763	TRUE	bleu-A	42.5
En-Ja	DLUT	789	TRUE	bleu-A	41.8
En-Ja	LanguageX	676	FALSE	bleu-A	41.7
En-Ja	JDExploreAcademy	516	TRUE	bleu-A	**41.5**
En-Ja	Lan-Bridge	555	FALSE	bleu-A	39.4
pair	**system**	**id**	**is_constrained**	**metric**	**score**
En-Ru	JDExploreAcademy	509	TRUE	bleu-A	**32.7**
En-Ru	Lan-Bridge	556	FALSE	bleu-A	32.6
En-Ru	HuaweiTSC	680	TRUE	bleu-A	30.8
En-Ru	PROMT	804	FALSE	bleu-A	30.6
En-Ru	SRPOL	265	TRUE	bleu-A	30.4
pair	**system**	**id**	**is_constrained**	**metric**	**score**
En-Zh	LanguageX	716	FALSE	bleu-A	54.3
En-Zh	HuaweiTSC	557	FALSE	bleu-A	49.7
En-Zh	JDExploreAcademy	834	TRUE	bleu-A	**49.7**
En-Zh	AISP-SJTU	611	TRUE	bleu-A	48.8
En-Zh	Manifold	336	TRUE	bleu-A	48.7
pair	**system**	**id**	**is_constrained**	**metric**	**score**
Cs-En	JDExploreAcademy	505	TRUE	bleu-B	**54.9**
Cs-En	Lan-Bridge	585	FALSE	bleu-B	54.5
Cs-En	CUNI-DocTransformer	805	TRUE	bleu-B	51.9
Cs-En	CUNI-Transformer	772	TRUE	bleu-B	51.6
Cs-En	SHOPLINE-PL	819	TRUE	bleu-B	46.8
pair	**system**	**id**	**is_constrained**	**metric**	**score**
De-En	JDExploreAcademy	809	TRUE	bleu-A	**33.7**
De-En	Lan-Bridge	587	FALSE	bleu-A	33.4
De-En	PROMT	796	FALSE	bleu-A	32.5
De-En	LT22	605	TRUE	bleu-A	26.0
pair	**system**	**id**	**is_constrained**	**metric**	**score**
Ja-En	NT5	766	TRUE	bleu-A	26.6
Ja-En	JDExploreAcademy	512	TRUE	bleu-A	**25.6**
Ja-En	DLUT	693	TRUE	bleu-A	24.8
Ja-En	Lan-Bridge	588	FALSE	bleu-A	22.8
Ja-En	NAIST-NICT-TIT	583	TRUE	bleu-A	22.7
pair	**system**	**id**	**is_constrained**	**metric**	**score**
Ru-En	Lan-Bridge	589	FALSE	bleu-A	45.2
Ru-En	HuaweiTSC	836	TRUE	bleu-A	45.1
Ru-En	JDExploreAcademy	769	TRUE	bleu-A	**45.1**
Ru-En	SRPOL	666	TRUE	bleu-A	43.6
Ru-En	ALMAnaCH-Inria	710	TRUE	bleu-A	30.3
pair	**system**	**id**	**is_constrained**	**metric**	**score**
Zh-En	JDExploreAcademy	708	TRUE	bleu-A	**33.5**
Zh-En	LanguageX	219	FALSE	bleu-A	31.9
Zh-En	HuaweiTSC	477	FALSE	bleu-A	29.8
Zh-En	AISP-SJTU	648	TRUE	bleu-A	29.7
Zh-En	Lan-Bridge	386	FALSE	bleu-A	28.1

Table 8: **Ranking of our submissions in terms of SacreBLEU-Score** in WMT2022 general translation task.

pair	**system**	**id**	**is_constrained**	**metric**	**score**
En-Cs	CUNI-Bergamot	734	TRUE	COMET-B	0.960
En-Cs	JDExploreAcademy	829	TRUE	COMET-B	**0.953**
En-Cs	Lan-Bridge	551	FALSE	COMET-B	0.947
En-Cs	CUNI-DocTransformer	800	TRUE	COMET-B	0.917
En-Cs	CUNI-Transformer	761	TRUE	COMET-B	0.866
pair	**system**	**id**	**is_constrained**	**metric**	**score**
En-De	JDExploreAcademy	843	TRUE	COMET-A	**0.632**
En-De	Lan-Bridge	549	FALSE	COMET-A	0.588
En-De	OpenNMT	207	FALSE	COMET-A	0.572
En-De	PROMT	694	FALSE	COMET-A	0.558
pair	**system**	**id**	**is_constrained**	**metric**	**score**
En-Ja	JDExploreAcademy	516	TRUE	COMET-A	**0.651**
En-Ja	NT5	763	TRUE	COMET-A	0.641
En-Ja	LanguageX	676	FALSE	COMET-A	0.621
En-Ja	DLUT	789	TRUE	COMET-A	0.605
En-Ja	Lan-Bridge	555	FALSE	COMET-A	0.565
pair	**system**	**id**	**is_constrained**	**metric**	**score**
En-Ru	JDExploreAcademy	509	TRUE	COMET-A	**0.696**
En-Ru	Lan-Bridge	556	FALSE	COMET-A	0.673
En-Ru	PROMT	804	FALSE	COMET-A	0.603
En-Ru	SRPOL	265	TRUE	COMET-A	0.597
En-Ru	HuaweiTSC	680	TRUE	COMET-A	0.592
pair	**system**	**id**	**is_constrained**	**metric**	**score**
En-Zh	LanguageX	716	FALSE	COMET-A	0.638
En-Zh	JDExploreAcademy	834	TRUE	COMET-A	**0.617**
En-Zh	Lan-Bridge	714	FALSE	COMET-A	0.614
En-Zh	Manifold	336	TRUE	COMET-A	0.601
En-Zh	HuaweiTSC	557	FALSE	COMET-A	0.595
pair	**system**	**id**	**is_constrained**	**metric**	**score**
Cs-En	JDExploreAcademy	505	TRUE	COMET-B	**0.747**
Cs-En	Lan-Bridge	585	FALSE	COMET-B	0.718
Cs-En	CUNI-DocTransformer	805	TRUE	COMET-B	0.706
Cs-En	CUNI-Transformer	772	TRUE	COMET-B	0.692
Cs-En	SHOPLINE-PL	819	TRUE	COMET-B	0.611
pair	**system**	**id**	**is_constrained**	**metric**	**score**
De-En	JDExploreAcademy	809	TRUE	COMET-A	**0.580**
De-En	Lan-Bridge	587	FALSE	COMET-A	0.565
De-En	PROMT	796	FALSE	COMET-A	0.518
De-En	LT22	605	TRUE	COMET-A	0.256
pair	**system**	**id**	**is_constrained**	**metric**	**score**
Ja-En	NT5	766	TRUE	COMET-A	0.420
Ja-En	JDExploreAcademy	512	TRUE	COMET-A	**0.406**
Ja-En	DLUT	693	TRUE	COMET-A	0.372
Ja-En	NAIST-NICT-TIT	583	TRUE	COMET-A	0.334
Ja-En	LanguageX	435	FALSE	COMET-A	0.329
pair	**system**	**id**	**is_constrained**	**metric**	**score**
Ru-En	JDExploreAcademy	769	TRUE	COMET-A	**0.649**
Ru-En	Lan-Bridge	589	FALSE	COMET-A	0.631
Ru-En	HuaweiTSC	836	TRUE	COMET-A	0.609
Ru-En	SRPOL	666	TRUE	COMET-A	0.595
Ru-En	ALMAnaCH-Inria	710	TRUE	COMET-A	0.268
pair	**system**	**id**	**is_constrained**	**metric**	**score**
Zh-En	JDExploreAcademy	708	TRUE	COMET-A	**0.451**
Zh-En	LanguageX	219	FALSE	COMET-A	0.449
Zh-En	Lan-Bridge	386	FALSE	COMET-A	0.430
Zh-En	HuaweiTSC	477	FALSE	COMET-A	0.428
Zh-En	AISP-SJTU	648	TRUE	COMET-A	0.416

Table 9: **Ranking of our submissions in terms of COMET-Score** in WMT2022 general translation task.

No Domain Left Behind

Hui Zeng
LanguageX AI Lab
felix_zeng_ai@aliyun.com

Abstract

We participated in the WMT General MT task and focus on four high resource language pairs: English to Chinese, Chinese to English, English to Japanese and Japanese to English). The submitted systems (LanguageX) focus on data cleaning, data selection, data mixing and TM-augmented NMT. Rules and multilingual language model are used for data filtering and data selection. In the automatic evaluation, our best submitted English to Chinese system achieved 54.3 BLEU score and 63.8 COMET score, which is the highest among all the submissions.

1 Introduction

Training neural machine translation models for a specific domain is a well-studied task. However, maximizing the performance of a single NMT model for multiple domains remains difficult. As a former translator and a current machine translation engineer, I always dream about building a versatile machine translation system – no domain left behind. Our neural machine translation system is developed using big transformer (Vaswani et al., 2017) architecture and the toolkit I used is fairseq (Ott et al., 2020). Rules, multilingual language model and faiss (Johnson et al., 2021) are used to align, clean and select parallel data. The following techniques are used in model training: a. Data mixing is used to mix general domain corpus with specific domain corpus; b. Back translation (Sennrich et al., 2016) is not applied because it is time-consuming. Instead, Neural Machine Translation with Monolingual Translation Memory (Cai et al., 2021) is used to fully utilize the monolingual corpus.

2 Data Filtering and Selection

The Chinese-English parallel data is mainly from CCMT Corpus[1], inhouse domain data from translation projects, as well as parallel data aligned from multilingual websites and e-books. The monolingual data for multiple domains is collected from the internet and e-books. WMT newstest2021 is used to evaluate the model's general domain performance. Multiple domain-specific test sets are created to evaluate the model's specific domain performance. Each domain has a test set of 1,000 sentences.

In order to build a versatile machine translation system, a total of 15 domains are covered in preparing the parallel and monolingual corpus. The primary domains and subdomains are listed as follows:

Literature
Web novel
Famous literary work
Literature/Poetry
Idioms/maxims/sayings
Slang
Conversation
Names (personal, company)
Symbols / Abbreviations / Acronyms

Art, History and Philosophy
Arts/crafts/painting
Cooking/culinary/gastronomy
Folklore
History
Philosophy
Graphic arts/photo/imaging
Music
Religion

[1] http://mteval.cipsc.org.cn:81/agreement/description

Social Science, Sociology, Ethics, etc.

Economy, Finance and Business

Business/commerce
Accounting
Finance (general)
Investment / Securities
Insurance
Economics
Real Estate

Fashion and Marketing

Advertising / Public Relations
Marketing / Market Research
Cosmetics / Beauty
Fashion
Textiles / Clothing / Fashion
Clothing/textiles

Politics and National Defense

Government/politics
International org/Dev/coop
Military / Defense

Law

Law (general)
Law: Contract(s)
Law: patents/trademarks/copyrights
Law: Taxation & Customs

Computers and IT (Information Technology)

Computers (general)
Computers: Systems, Networks
Computers: Hardware
IT (Information Technology)
Telecommunications
Internet, e-Commerce
SAP System Applications and Products
Media / Multimedia

Films and Television

Cinema/film/TV/drama

Games, Sports and Entertainment

Games / Video Games / Gaming / Casino
Sports / Fitness / Recreation
Tourism & Travel

Medical

Medical (general)
Medical: Cardiology
Medical: Dentistry
Medical: Health Care
Medical: Instruments
Medical: Pharmaceuticals
Dentistry
Veterinary
Genetics
Nutrition

Industry and Engineering

Engineering
Nuclear Eng/Sci
Automation & Robotics
Automotive / Cars & Trucks
Mechanics / Mech Engineering
Construction / Civil Engineering
Transport / Transportation / Shipping
Electronics / Elect Eng
Petroleum Eng/Sci
Surveying
Metallurgy / Casting
Mining & Minerals / Gems
Energy / Power Generation
Maritime / Sailing / Ships
Industrial
Food/drink
Paper / Paper Manufacturing
Printing & Publishing
Nuclear
Manufacturing
Furniture/household/appliance
Materials (Plastics, Ceramics, Rubber, Glass, Wood etc.)

Science

Astronomy/space
Aerospace/aviation/space
Mathematics & Statistics
Physics
Chemistry
Geography/geology
Architecture
Zoology
Biology
Botany
Meteorology
Metrology
Psychology
Education/pedagogy
Linguistics
Environment & Ecology
Anthropology
Archaeology
Genealogy

Agriculture and Animal Husbandry

Agriculture
Fisheries
Forestry wood timber
Wood Industry = Forestry
Wine / Oenology / Viticulture
Animal husbandry/livestock

Management and Training

Management

Human Resources
Safety

News and Journalism

2.1 Monolingual Data Filtering

The monolingual data for 15 primary domains are mainly collected from websites and e-books. The following rules are used for a simple cleaning:
•Remove duplicated sentences.
•Remove the sentences containing special characters.
•Remove the sentences containing html addresses or tags.

2.2 Parallel Corpus Aligning

There are a large number of multilingual websites and multilingual e-books, which are easily accessible. However, these data need to be aligned to create sentence level parallel corpus. To this end, a corpus aligner is created using Sentence-BERT (Reimers et al., 2019) and faiss (Johnson et al., 2021).

Regardless of order, thousands of source sentences and target sentences are first encoded into sentence embeddings using Sentence-BERT, and then faiss is used to retrieve the target sentences which is most similar in meaning to the source sentences. The aligning of thousands of parallel sentences could be finished within a few seconds.

2.3 Parallel Data Filtering Using Rules

The following rules are used to filter parallel corpus.

a. Remove duplicated sentence pairs.
b. Remove the lines having identical source and target sentences.
c. Remove the sentence pairs containing special characters.
d. Remove the sentence pairs containing html addresses or tags.
e. Remove the sentence pairs with empty source or target side.

2.4 Parallel Data Filtering Using Multilingual Language Model

As mentioned in section 2.2, a corpus aligner is created using Sentence-BERT (Reimers et al., 2019) and faiss (Johnson et al., 2021). This can also be used to filter parallel data.

Apart from the corpus aligned from websites and e-books, in-house data from translation projects and public corpus like CCMT are also used.

The aforesaid corpus aligner can be used to score each parallel sentence pair so that the pairs with extremely low scores can be removed.

3 System Description

This section illustrates how the model is trained step by step.

3.1 Data pre-processing

For data preprocessing, we use the tokenizer developed on my own to process both Chinese and English. Chinese text (including punctuations and numbers) is split to single character level. We keep the upper- and lower-case letters of English as they are, since we believe they are also important features for the model. Numbers in English text are also split into single digits. We use byte pair encoding (BPE) (Sennrich et al., 2016) to create a shared vocabulary, so that the vocabulary size is reduced to 45467. We also wrote a post-processor to restore the Chinese and English text to normal form.

3.2 Baseline Model Training

WMT newstest2021 is used to evaluate the model's general domain performance. Multiple domain-specific test sets are created to evaluate the model's specific domain performance. Each domain has a test set of 1,000 sentences.

The CCMT parallel Corpus filtered by rules and corpus aligner is used to train big transformer (Vaswani et al., 2017) English to Chinese and Chinese to English translation models as the general domain baselines.

Validation is performed every 2000 steps. The training is terminated if there is no gain in BLEU (Papineni et al., 2002) for 20 consecutive validations.

The BLEU scores on specific domains are also calculated as baselines.

3.3 Training on Mixed Data

Data mixing (Hasler et al., 2021) is used to improve translation quality for multiple new domains represented by small amounts of parallel data while maintaining the performance of a high-quality, general-purpose NMT model.

The importance of the training data sample can be increased by increasing its size, thereby

Model + Corpus	Literature EN2ZH	Law EN2ZH	Medical EN2ZH	Newstest2021 EN2ZH
filtered CCMT Corpus big transformer	21.5	23.2	19.8	28.7
filtered CCMT Corpus data mixing (general data, domain specific data) big transformer	28.7	38.6	36.3	35.9
filtered CCMT Corpus data mixing (general data, domain specific data) NMT with domain specific monolingual translation memory	31.2	43.5	40.1	39.2

Table 1: Different systems and their BLEU scores (only three typical domains are listed)

changing the ratio of training data and domain data to influence the trade-off between generic and domain performance.

3.4 NMT with Monolingual Translation Memory

Prior work has proved that Translation memory (TM) can boost the performance of Neural Machine Translation (NMT). In contrast to existing work that uses bilingual corpus as TM and employs source-side similarity search for memory retrieval, Cai (Cai et al., 2021) proposed a new framework that uses monolingual memory and performs learnable memory retrieval in a crosslingual manner.

This framework has unique advantages. First, the cross-lingual memory retriever allows abundant monolingual data to be TM. Second, the memory retriever and NMT model can be jointly optimized for the ultimate translation goal. The "plug and play" property of TM is useful for domain adaptation, where a single general-domain model can be adapted to a specific domain by using domain-specific monolingual TM.

3.5 Results

The BLEU scores on general test sets and some domain specific test sets for each corpus plus model combination are shown in Table 1.

In the automatic evaluation, our best submitted English to Chinese system achieved 54.3 BLEU score and 63.8 COMET score, which is the highest among all the submissions.

4 Conclusion

This paper describes LanguageX's translation system for the WMT2022 General MT task. The potential of a single translation model for all domains is explored. We are pleased to argue that, with data mixing and TM-augmented NMT, a versatile machine translation system with all-round translation performance could be built.

Acknowledgments

Thanks to my wife who spends most of her time to take care of our two kids, so that I am able to participate in the contest and complete this paper.

References

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *In Proceedings of the 31st International Conference on Neural Information Processing Systems,* pages 6000–6010.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-Scale Similarity Search with GPUs. *In IEEE Transactions on Big Data*, pp. 535-547, vol. 7.

Deng Cai, Yan Wang, Huayang Li, Wai Lam, and Lemao Liu. 2021. Neural Machine Translation with Monolingual Translation Memory. *In*

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 7307–7318 August 1–6, 2021. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 3982–3992, Hong Kong, China, November 3–7, 2019*. Association for Computational Linguistics

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. *In Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* pages 86–96. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Eva Hasler, Tobias Domhan, Jonay Trenous, Ke Tran, Bill Byrne, and Felix Hieber. 2021. Improving the Quality Trade-Off for Neural Machine Translation Multi-Domain Adaptation. *In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing,* pages 8470–8477, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

GTCOM Neural Machine Translation Systems for WMT22

Hao Zong, Chao Bei, Conghu Yuan
Global Tone Communication Technology Co., Ltd
{zonghao,yuanconghu}@gtcom.com.cn
chaobei001@gmail.com

Abstract

This paper describes the Global Tone Communication Co., Ltd.'s submission of the WMT22 shared general MT task. We participate in six directions: English to/from Ukrainian, Ukrainian to/from Czech, English to Croatian and English to Chinese. Our submitted systems are unconstrained and focus on backtranslation, multilingual translation model and finetuning. Multilingual translation model focus on X to one and one to X. We also apply rules and language model to filter monolingual, parallel sentences and synthetic sentences.

1 Introduction

We applied fairseq(Ott et al., 2019) as our develop tool and use transformer(Vaswani et al., 2017) as the main architecture. The primary ranking index for submitted systems is BLEU(Papineni et al., 2002), therefore we apply BLEU as the evaluation matrix for our translation system by using sacreBLEU[1].

For data preprocessing, punctuation normalization, tokenization and BPE(byte pair encoding) are applied for all languages. Further, we apply truecase model for English, Ukrainian, Czech and Croatian according to the character of each language. Regarding to the tokenization, we use polyglot as the tokenizer for Ukrainian and Croatian, and mosese tokenizer.perl for English and Czech. Besides, knowledge based rules and language model are also involved to clean parallel data, monolingual data and synthetic data.

This paper is arranged as follows. We firstly describe the task and show the data information, then introduce our baseline and multilingual translation model. After that, we describe the conducted experiments in detail in all directions, including data preprocessing, model architecture, back-translation and multilingual translation model. At last, we analyze the results of experiments and draw the conclusion.

2 Task Description

The task focuses on bilingual text translation and the provided data is shown in Table 1, including parallel data and monolingual data. For the directions between English and Ukrainian, the parallel data is mainly from ParaCrawl v9, WikiMatrix, Tilde MODEL corpus and OPUS, as well as the directions English to Croatian. For the directions between Ukrainian and Czech, the parallel data is mainly from WikiMatrix and OPUS. The monolingual data we used includes: News Crawl in English, Ukrainian, Croatian and Czech; Leipzig Corpora in Croatian, Ukrainian and Czech; News discussions in English. All language directions we participated in are new tasks this year, therefore we only use the provided development set from FLoRes101 dataset for all directions.

Usually, the news translation task will take the human evaluation result as the final ranking index. And this requires each participated team contribute 8 hours of human evaluation for each participating translation direction. For some low resource language directions, it is not very easy for the organizer to employ human translators from the participating team or translation agency. Besides, due to the number of sentences in the test set and the quantity of participating teams, it is not possible to employ human evaluation for all the test sets. Besides, with recent improvements of MT quality, the organizer decided to move away from testing only in the news domain and we are shifting the WMT focus on testing the general capabilities of MT systems.

3 Billingual Baseline Model and Multilingual Translation Model

To set a strong baseline for our multilingual model as a comparison. Our Billingual base-

[1] https://github.com/mjpost/sacrebleu

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 428 - 431
December 7-8, 2022 ©2022 Association for Computational Linguistics

language	number of sentences
en-hr parallel data	318M
en-uk parallel data	13M
uk-cs parallel data	4M
en monolingual data	40M
uk monolingual data	15M
cs monolingual data	40M
hr monolingual data	13M
en-uk development set	997
en-hr development set	997
uk-cs development set	997

Table 1: Task Description

line model is different from the transformer base model transformer_wmt_en_de with 6 encoding layers and 6 decoding layers. Instead, we set our bilingual baseline model by using transformer_vaswani_wmt_en_de_big architecture with 12 encoding layers and 4 decoding layers.

The multilingual translation model is almost the same as GTCOM2021(Bei and Zong, 2021), but focuses on one to X and X to one this year. To obtain a better translation quality, we include Russian as the main auxiliary language since Russian and Ukrainian are very similar. We train four multilingual models: 1. ru-en, uk-en and hr-en to translate uk-en; 2. en-ru, en-uk and en-hr to translate en-uk and en-hr; 3. cs-uk, en-uk and ru-uk to translate cs-uk 4. en-uk; uk-cs and en-cs to translate uk-cs and en-cs. We use joint BPE for all languages in the multilingual model separately.

For English to Chinese direction, we just test our online system as a comparison with other participating systems. Therefore we did not conduct data augmentation, finetuning, or any other adaption experiments.

4 Experiment

4.1 Training Step

This section introduces all the experiments we set step by step and Figure 1 shows the whole flow.

- **Date Filtering** The methods of data filtering are mainly the same as we did last year, including human rules, language models, and repeat cleaning.

- **Baseline.** We use big transformer architecture with 24 layers of encoder and 4 layers of decoder to construct our baseline.

model	en2uk	uk2en
baseline	32.43	40.08
back translation	32.58	40.84
joint training	32.97	42.33
deep multilingual translation model	33.72	43.27

Table 2: The BLEU score between English and Ukrainian.

model	uk2cs	cs2uk
baseline	22.52	22.00
back translation	25.51	23.59
joint training	25.72	24.09
deep multilingual translation model	26.14	24.89

Table 3: The BLEU score between Czech and Ukrainian.

- **Back-translation.** We use a multilingual translation model to translate the target sentence to the source side, and clean synthetic data with language model. Here, we translate each language pairs we have added into the multilingual translation model. Mix cleaned back-translation data and parallel sentences and train multilingual translation model.

- **Joint training.** Repeat the back-translation step by the best model, until there is no improvement.

- **Multilingual translation model.** We focus on one to X and X to one model, and each multilingual model has joint BPE and a shared vocabulary. The multilingual translation model setting follows Google's Multilingual Neural Machine Translation System(Johnson et al., 2017).

- **Deep multilingual translation model.** Using bilingual parallel data and synthetic data by the best model, train the multilingual transformer model with 12 encoding layers and 4 decoding layers, then repeat the back-translation step and forward-translation step, until there is no improvement.

- **Ensemble Decoding.** We use GMSE Algorithm (Deng et al., 2018) to select models to obtain the best performance.

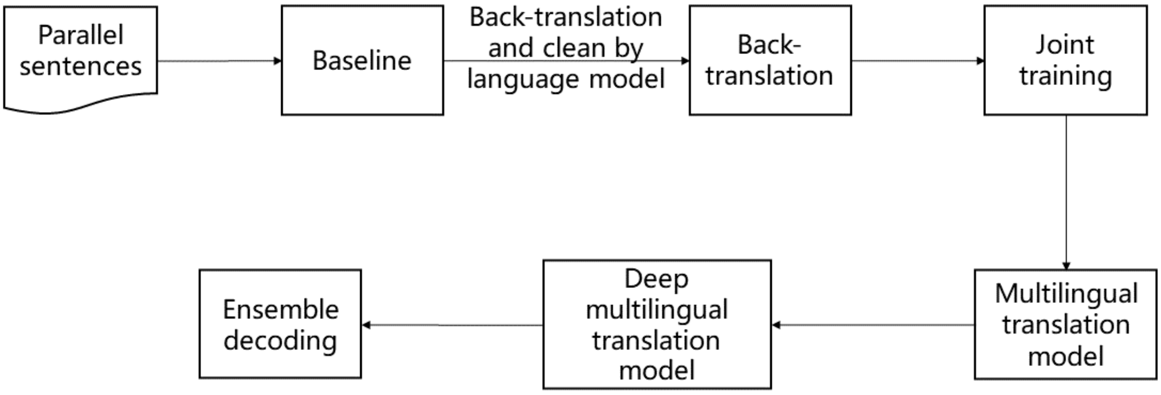

Parallel sentences
Baseline
Back-translation and clean by language model
Back-translation
Joint training
Multilingual translation model
Deep multilingual translation model
Ensemble decoding

Figure 1: The work flow of GTCOM machine translation competition systems

model	en2hr
baseline	30.15
back translation	32.90
joint training	33.80
deep multilingual translation model	34.93

Table 4: The BLEU score for English to Croatian.

5 Result and Analysis

Table 2, Table 3 and Table 4 show the BLEU score we evaluated on development set for English to/from Ukrainian, Czech to/from Ukrainian and English to Croatian respectively. As shown in the above table, back-translation is still the best data augmentation measure to improve translation quality from the data aspect. Joint training and deep multilingual translation model also show solid improvement in all five directions.

We notice that when adding Russian (a very similar language to Ukrainian) into the multilingual corpus, we did not obtain as much improvement as we expect. This is probably because the original English to Ukrainian data is rich enough and decreased the positive impact of adding Russian data into the multilingual model.

6 Conclusion

This paper describes GTCOM's neural machine translation systems for the WMT22 shared general MT task. We applied 3 major techniques to improve the translation quality: back-translation, joint training, and deep multilingual translation model. With these 3 techniques, the final automatic evaluation matrix is shown in Table 5. Besides BLEU, this year the organizer introduce a new evaluation matrix COMET(Rei et al., 2020) to inspect the translation quality. Our system is ranking 1st place in English->Ukrainian and English->Chinese, 2nd place in Ukrainian-English, Czech ->Ukrainian and English->Croatian, 7th place in Ukrainian->Czech with COMET index.

Direction	BLEU	COMET Rank
en2uk	30.8	1
uk2en	43.9	2
cs-uk	36.8	2
uk-cs	31.3	7
en-hr	17.6	2
en-zh	47.7	1

Table 5: The final online automatic evaluation result.

Acknowledgments

The authors gratefully acknowledge the financial support provided by the National Key Research and Development Program of China (2020AAA0108005). And this work is supported by 2030 Aritificial Intellegence Research Institute of Global Tone Communication Technology Co., Ltd.[2]

References

Chao Bei and Hao Zong. 2021. GTCOM neural machine translation systems for WMT21. In *Proceedings of the Sixth Conference on Machine Translation*, pages 100–103, Online. Association for Computational Linguistics.

[2] https://www.gtcom.com.cn

Yongchao Deng, Shanbo Cheng, Jun Lu, Kai Song, Jingang Wang, Shenglan Wu, Liang Yao, Guchun Zhang, Haibo Zhang, Pei Zhang, et al. 2018. Alibaba's neural machine translation systems for wmt18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 368–376.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Linguistically motivated Evaluation of the 2022 State-of-the-art Machine Translation Systems for three Language Directions

Vivien Macketanz[1], Shushen Manakhimova[1], Eleftherios Avramidis[1],
Ekaterina Lapshinova-Koltunski[2], Sergei Bagdasarov[3] and Sebastian Möller[1]

[1]German Research Center for Artificial Intelligence (DFKI)
firstname.lastname@dfki.de
[2]University of Hildesheim, lapshinovakoltun@uni-hildesheim.de
[3]Saarland University, s8sebagd@stud.uni-saarland.de

Abstract

This document describes a fine-grained linguistically motivated analysis of 29 machine translation systems submitted at the Shared Task of the 7th Conference of Machine Translation (WMT22). This submission expands the test suite work of previous years by adding the language direction of English–Russian. As a result, evaluation takes place for the language directions of German–English, English–German, and English–Russian. We find that the German–English systems suffer in translating idioms, some tenses of modal verbs, and resultative predicates, the English–German ones in idioms, transitive-past progressive, and middle voice, whereas the English–Russian ones in pseudogapping and idioms.

1 Introduction

Neural Machine Translation has seen enormous progress and reached a quality that is helpful for many everyday use cases. However, several methods for evaluating MT suggest that there is still plenty of room for improvement. An evaluation method for revealing the translation flaws in a more structured way refers to the use of *test suites* or *challenge sets*. Contrary to the classical evaluation, where test sets are drawn from random everyday texts, test suites consist of manually devised or selected sentences that focus on testing the ability of the MT systems to translate a particular phenomenon. Here, we are presenting test suite results while analyzing the state-of-the-art systems with regard to many linguistically-motivated phenomena. The test suites[1] were applied to the MT systems submitted at the 7th Conference of Machine Translation (WMT22) for the language directions German–English, English–German, and English–Russian. The test suites for the first two language directions have also been showcased during the previous years, whereas English–Russian is published for the first time.

This paper is structured as follows: Section 2 goes through related papers, whereas Section 3 explains how the test suite was created and applied. Section 4 outlines the setup of this year's experiment, whose results are detailed in Section 5. Section 6 concludes the paper with an outlook to future research.

2 Related Work

The first test suites were introduced as early as the first MT systems in the 1990s (King and Falkedal, 1990; Way, 1991; Heid and Hildenbrand, 1991). Recent years saw the rise of Deep Learning and the drastic improvement of the quality of MT outputs, which has led to the current revival of test suites. Most of these test suites, however, focus on evaluating specific linguistic phenomena, e.g., Guillou and Hardmeier (2016), or on the comparison of different MT technologies (Isabelle et al., 2017; Burchardt et al., 2017), and Quality Estimation methods (Avramidis et al., 2018).

Over the last few years, several test suites for multiple language directions have emerged as a part of the Conference on Machine Translation test suite track. These test suites, however, focus on one or a few different phenomena, including the works of Popović (2019) Cinkova and Bojar (2018), Bojar et al. (2018), Rysová et al. (2019), Vojtěchová et al. (2019), Kocmi et al. (2020), Zouhar et al. (2020), Burlot et al. (2018), Guillou et al. (2018), Rios et al. (2018), Raganato et al. (2019), Scherrer et al. (2020). Our test suite, on the other hand, performs a systematic evaluation of more than one hundred phenomena per language direction (Macketanz et al., 2022). Similar to our work, the test suite approach and human evaluation are also used to evaluate MT quality metrics (Freitag et al., 2021; Avramidis and Macketanz, 2022).

[1] https://github.com/DFKI-NLP/mt-testsuite

Test set	Test sentences	Categories	Phenomena
De–En	~5,500	14	106
En–De	~4,400	13	110
En–Ru	~300	12	51

Table 1: Metadata of the language pairs in the test suite.

3 Method

We have created a large-scale test suite with the goal of testing and comparing the performance of MT systems. Currently, the test suite covers four different language pairs. We will present three in this paper: German to English, English to German, and English to Russian (the fourth language pair being Portuguese to English). The test suite is based on a number of linguistic categories which are in turn divided into more fine-grained linguistic phenomena. The categories and phenomena are language-specific; however, there is a significant overlap between many of the categories and phenomena across the different language pairs. Each linguistic phenomenon in the test suite is represented by multiple test sentences. All categories, phenomena, and test sentences are the result of extensive research and knowledge of the syntax and morphology of the languages under inspection. The categories and phenomena do not follow a specific linguistic theory, but they were created by linguistic experts who are native speakers or highly proficient speakers of the languages. Furthermore, the set of categories and phenomena was reviewed internally by linguists and experienced translators to achieve objectivity in the classification.

The number of test sentences, categories, and phenomena for each language pair can be found in Table 1. As can be seen in the table, the English–Russian test set is considerably smaller than the other two test sets. This is due to the fact that we started creating the English–Russian test set only recently. However, we are currently working on expanding the test set by creating more phenomena and test sentences.

In order to allow for a semi-automatic evaluation of the test sentences, we have created a set of rules which determine whether a test sentence is translated correctly or incorrectly. The rules consist of hand-crafted regular expressions and fixed strings of translation outputs. They can be applied with the help of an internal evaluation tool (Macketanz et al., 2022). The workflow of the preparation and application of the test suite is depicted in Figure 1.

3.1 Application of the test suite

The thorough building and application of the test suite can be found in the previous test suite track papers (Macketanz et al., 2018; Avramidis et al., 2019, 2020; Macketanz et al., 2021). This paper gives a quick overview of the whole system. As shown in Figure 1, the building of the test suite follows steps a to c. Once the test sentences are fed as input to the MT systems, begins the application of the test suite (step d). The MT outputs are then automatically evaluated by the test suite tool with the help of the rules defined earlier by linguists and annotators (step e). The rules combine pre-set regular expressions and fixed strings (correct and incorrect translations from earlier MT system outputs). The rules are designed to evaluate each phenomenon in question's correct and incorrect translations. Note that only the phenomenon under inspection is being evaluated, meaning that all translation errors that are unrelated to the phenomenon are being ignored. The test sentence is marked with a warning if the output cannot be automatically sorted as correct or incorrect with the predefined rules. These warnings are then manually reviewed by human linguist annotators who decide on the translation's correctness and adapt the rules accordingly (step e). After that, the phenomenon-specific translation accuracy is calculated by dividing the number of correctly translated test sentences of a phenomenon by the total number of test sentences of that phenomenon:

$$\text{accuracy} = \frac{\text{correct translations}}{\text{sum of test items}}$$

Since this evaluation aims to compare the systems fairly, only the test items that do not contain any warnings for any systems are included in the calculation. If a test item has an unresolved warning for any MT systems, we exclude them from the calculation. Unfortunately, this reduces the number of test items. We see great importance in the extensive manual evaluation and human annotators designing rules with good coverage.

To define which system(s) perform better for a particular phenomenon (or category), we first identify the best scoring system in each language direction and then compare it to other systems. To do so, we confirm the significance of the comparison with a one-tailed Z-test with $\alpha = 0.95$. The systems that do not differ significantly from the best system are considered in the first performance cluster and indicated with boldface in the tables.

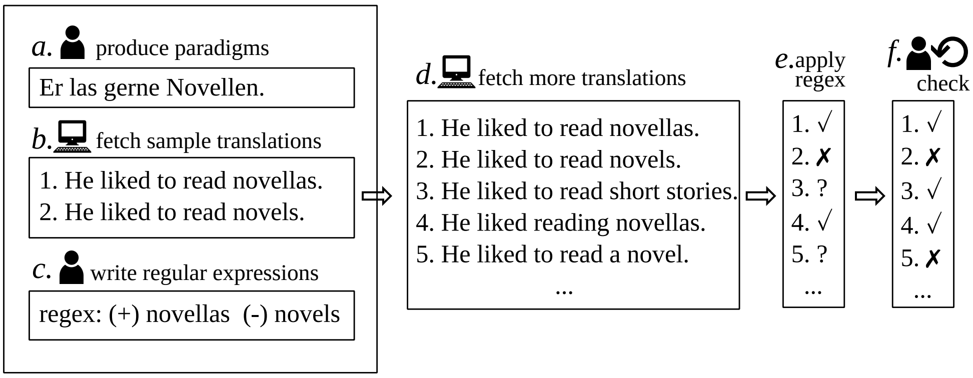

a. produce paradigms
Er las gerne Novellen.
b. fetch sample translations
1. He liked to read novellas.
2. He liked to read novels.
c. write regular expressions
regex: (+) novellas (-) novels
d. fetch more translations
1. He liked to read novellas.
2. He liked to read novels.
3. He liked to read short stories.
4. He liked reading novellas.
5. He liked to read a novel.
...
e. apply regex
1. ✓
2. ✗
3. ?
4. ✓
5. ?
...
f. check
1. ✓
2. ✗
3. ✓
4. ✓
5. ✗
...

Figure 1: Example of the preparation and application of the test suite for one test sentence

The boldfaces, therefore, have a meaning only for the respective row of the table.

The average scores are computed in three ways as each category or phenomenon has a different number of test items. Micro-average aggregates the contributions of all test items to compute the average percentages. Category macro-average computes the percentages independently for each category and then averages them (i.e., treating all categories equally). Phenomenon macro-average computes the percentages independently for each phenomenon and then takes the average (i.e., treating all phenomena equally).

4 Experiment Setup

In this paper, we present the evaluation of 29 systems with our test suite. The systems are part of the *news translation task* of the Seventh Conference on Machine Translation (WMT22). The systems cover three different language pairs: nine systems for German–English, nine systems for English–German, and 11 systems for English–Russian.

This year is the second time that the English–German systems are being evaluated and the first time that the English–Russian systems are being evaluated with our test suite. Every year, manual work is involved upon receiving the system translations as there are usually a number of translation outputs that are not yet covered by the existing rules in the database (the warnings). This year, there were on average 7.8 % of warnings for German–English, 9.7 % for English–German, and 20,6 % for English–Russian. It is not surprising that the English–Russian test set had a comparably bigger amount of warnings as this was the first time the test set was evaluated and therefore, the database of evaluation rules for this language pair was still rather small. It was also expected that English–German would have a higher amount of warnings than German–English as the German–English test set has the largest rules database since this language pair has been evaluated five years in a row.

Two annotators with extensive linguistic knowledge of the three languages under investigation conducted the manual evaluation of the warnings. No inter-annotator agreement was calculated; however, problematic cases were discussed with several linguistic experts to exclude subjectivity. The manual evaluation took around four weeks and involved around 50 person-hours. After the manual evaluation, there were on average 1.2 % of warnings left for German–English, 3.2 % for English–German, and 0.7 % for English-Russian.

As mentioned above, test sentences with at least one warning by one system were excluded from the analysis to achieve a fair comparison between the systems under inspection. As a result, our analysis was conducted on 5049 (91 %) test sentences for German–English, 3723 (83 %) test sentences for English–German, and 300 (97 %) test sentences for English–Russian.

5 Results

All result tables can be found in the Appendix.

5.1 System comparison

For **German–English**, two systems have the highest micro-average (85 %), Online-W and Online-A, whereas when considering the macro-average, three more systems also achieve the highest scores (89-90 %), Online-B, Land-Bridge, and JDExplore-Academy.

For **English–German**, two systems have the highest micro-average (97 %), Online-B and Lan-

Bridge. However, on the macro-average, a different system displays the highest score (94 %), JDExploreAcademy. The system with the lowest micro- and macro-average, Online-Y, still achieves scores of 84 % for both averages.

For **English–Russian**, the same four systems achieve the highest scores on both the micro- (78-81 %) and the macro-average (82-85 %), Online-W, Online-G, Online-B, and JDExploreAcademy. The average scores of English–Russian on the category level are comparably smaller than the scores of German–English and English–German. One plausible explanation is that English and Russian are more distant from a typological perspective than English and German.

5.2 Category-level analysis

For **German–English**, the categories with the highest average by all systems (> 90 %) are *composition*, *coordination & ellipses*, *named entity & terminology*, *negation*, and *non-verbal agreement*. The category with the lowest average score (77.2 %) is *false friends*.

For **English–German**, the categories with the highest average scores (> 96 %) are *function words*, *negation*, *non-verbal agreement*, *subordination*, and *verb tense/aspect/mood*. The category with the lowest average score (77.8 %) is *punctuation*.

For **English–Russian**, the category with the highest average score (92 %) is *punctuation*, with seven of the 11 systems achieving 100 % of accuracy, followed by *ambiguity*, *function words*, *negation*, and *subordination* (all > 80 %). The category with the lowest accuracies is *coordination & ellipsis*, followed by *false friends*.

5.3 Phenomenon-level analysis

For **German–English**, there are many phenomena that reach an average of 90-100 %, while the phenomenon macro-average reaches 85 %. Phenomena that reach more than 95 % of accuracy are *gapping*, *sluicing*, *polar question*, *verbal MWE*, *date*, *measuring unit*, *negation*, *internal possessor*, *comma*, *infinitive clause*, *object clause*, several verb tenses in *ditransitive*, *intransitive*, *transitive*, and *modal verbs*, and *passive voice*.

Yet there are some phenomena with a very low accuracy: The phenomena *idiom*, *modal pluperfect*, *modal pluperfect subjunctive II modal negated pluperfect*, *modal negated pluperfect subjunctive II*, and *resultative predicates* are the phenomena with the lowest averages, ranging only between 20-57 % accuracy. This result goes hand in hand with last year's result where the phenomena *modal pluperfect*, *resultative predicates*, and *idioms* reached the lowest accuracy.

Idiom	
Er macht aus einer Mücke immer gleich einen Elefanten.	
It always makes out of a mosquito an elephant.	fail
He always turns a gnat into an elephant.	fail
He always makes a mountain out of a molehill.	pass
Modal negated pluperfect	
Ich hatte nicht lesen sollen.	
I wasn't supposed to read.	fail
I shouldn't have read.	fail
I didn't want to read.	fail
Right node raising	
Lena soll und Tim will den Vertrag kündigen.	
Lena will and Tim will terminate the contract.	fail
L. should and T. want to terminate the contract.	fail
L. should and T. wants to terminate the contract.	pass

Table 2: Examples of German–English linguistic phenomena with passing and failing MT outputs.

Table 2 contains example outputs from three different phenomena for German–English. The first example is from the phenomenon *idiom*. Idioms are multiword expressions whose meaning goes beyond the meaning of their separate elements. This also means that a simple literal translation into another language is usually incorrect. In our example at hand, the German idiom "aus einer Mücke einen Elefanten machen" means "to blow something out of proportion". A literal translation like the first and second outputs leads to an incorrect English meaning. What is further interesting about the incorrect outputs is that while the second one ("turns a gnat into an elephant") is at least grammatically correct, the first one ("makes out of a mosquito an elephant") is also grammatically incorrect. The translation of "Mücke" ("mosquito") as the term "gnat" is also unexpected. Only the third translation "makes a mountain out of a molehill" is a correct translation of this idiom.

The second example contains a *negated modal verb* in the *pluperfect tense*. The German sentence "Ich hatte nicht lesen sollen." can only be correctly translated as "I had not been supposed to read". This year, all systems failed to produce this correct output. Instead, there were different incorrect outputs with incorrect tenses ("I wasn't supposed to read.", "I shouldn't have read.") or incorrect translations of the modal verb ("I didn't want to read.").

The third example sentence contains an elliptical *right node raising construction*. *Right node raising constructions* often consist of parallel coordinate sentences (sentences joined by “and”) in which two conjuncts share some material on the right side of the structure. In the example sentence, the two conjuncts “Lena soll” (“Lena should”) and “Tim will” (“Tim wants to”) are sharing the material “den Vertrag kündigen” (“terminate the contract”) on the right side of the construction. In the first incorrect example, the verbs “soll” and “will” are both translated as “will” which is an incorrect translation for both verbs. In the second incorrect output, the verbs are translated correctly, however, the verb “want” is incorrectly conjugated, missing the third person singular ending. Surprisingly, there were multiple systems that created this incorrectly conjugated translation.

At this point, it is also interesting to mention that there was one system that often incorrectly conjugated the verb “to sleep” in the past tense: Instead of “slept”, the outputs by that particular system often contained the non-existing conjugation “sleeped”.

For **English–German**, the phenomenon-level macro-average is similarly high as for the other language direction with 93 %. The phenomena for which all systems reach 100 % accuracy are *question tag*, *compound*, *prepositional MWE*, *subject clause*, *intransitive - present perfect progressive, present perfect simple, simple present*, and *transitive - future I progressive*.

The phenomena with the lowest accuracies, ranging between 35-61 %, are *idioms*, *transitive - past progressive*, and *middle voice*. These results are more in line with last year’s results, as *idioms* and *middle voice* were also among the lowest accuracy phenomena.

Table 3 contains correct and incorrect translation examples from English–German. The first example contains a *coreference*. While many English nouns are gender-neutral, the same German nouns are in most cases gender specific. This can lead to translation errors if the context of a sentence clarifies the gender in English yet the German translation contains the incorrect gender. The test sentence at hand provides a clear context of the nurse being male. Yet, many systems incorrectly translated “nurse” as the female “Krankenschwester’ instead of the male “Krankenpfleger”. [2]

Coreference	
My brother is a nurse in the local hospital.	
Mein Bruder ist Krankenschwester im örtlichen Krankenhaus.	fail
Mein Bruder ist Krankenpfleger im örtlichen Krankenhaus.	pass
Verbal MWE	
She takes after her mother.	
Sie nimmt nach ihrer Mutter.	fail
Sie hinterlässt ihre Mutter.	fail
Sie kommt nach ihrer Mutter.	pass
Transitive future II progressive	
I will have been playing the piano.	
Ich würde Klavier gespielt haben.	fail
Ich habe Klavier gespielt.	fail
Ich werde Klavier gespielt haben.	pass

Table 3: Examples of English-German linguistic phenomena with passing and failing MT outputs.

The second example contains the *verbal multiword expression* “to take after somebody”. As explained above, multiword expressions cannot be translated literally as their meaning goes beyond their separate elements. The first incorrect output “Sie nimmt nach ihrer Mutter.” is, however, a literal translation of this multiword expression. The second incorrect output “Sie hinterlässt ihre Mutter.” is not a literal translation, yet still incorrect as it means “She leaves her mother behind”. Only the translation “Sie kommt nach ihrer Mutter.”, which is the German equivalent of this multiword expression, is correct.

The third example output contains a *transitive verb* in the tense *future II progressive*. The future II tense was often mistranslated as a conditional II tense “würde gespielt haben” (“would have played”) instead of the correct form “werde”. The second incorrect output contains a completely incorrect tense, the present perfect “habe gespielt” (“have played”).

For **English–Russian**, the phenomenon level macro-average accuracy lies at 76 %. Also for this language pair, there are some phenomena which reach 100 % accuracy for all systems, like *nominal MWE*, *prepositional MWE*, *contact clause*, *indirect speech*, and passive voice. On the other hand, there are quite a few phenomena that reach a very low accuracy, ranging between 30-50 %: *gapping*, *pseudogapping*, *idioms*, *verbal MWE*, *anaphora agreement*, *intransitive verbs*, and *middle voice*. The low accuracies of *idioms* and *verbal MWEs* are

[2] We are aware that genders and their translation are a large topic on their own which we can only scratch on the surface within the scope of our test suite. We would like to point the interested reader to the following research: (Hardmeier et al., 2022)

Collocation	
She is careful to eat light and exercise often.	
Она старается есть легкую пищу и часто занимается спортом.	pass
Она старается есть свет и часто тренируется.	fail
Она осторожно ест легко и часто занимается спортом.	fail
Она следит за тем, чтобы есть мало и часто заниматься спортом.	pass
Pseudogapping	
I don’t know that and don’t think you do.	
Я этого не знаю и не думаю, что вы это делаете.	fail
Я этого не знаю и не думаю, что знаешь.	fail
Я не знаю этого и не думаю, что вы знаете.	pass
Resultative	
He read the children to sleep.	
Он зачитывал детей спать.	fail
Он читал детям спать.	fail
Он читал детям перед сном.	pass

Table 4: Examples of English–Russian linguistic phenomena with passing and failing MT outputs.

not surprising as multiword expressions generally tend to cause translation errors across all language pairs. What is interesting is that the accuracy of *intransitive verbs* is considerably lower than the accuracies of the other verb types. One potential reason might be that in our small-scale English–Russian test suite the intransitives are presented by the verb of motion “to go”, which has a number of equivalents in Russian that can convey various aspects such as tense, frequency, or incompleteness. This ambiguity increases the overall number of equivalents in the training data which could lead to faulty results when analyzing the translations with respect to specific phenomena.

Table 4 covers example translations of some low-accuracy phenomena for English–Russian. The first example contains the *collocation* “to eat light” that does not have an exact equivalent in Russian. The word “light” poses some extra difficulty, as it is lexically and semantically ambiguous in both languages. In different contexts, it could function as an adverb, adjective, or noun. This year, a typical incorrect output is “есть свет” (*est’ svet*) meaning to consume light as in electromagnetic radiation, and “есть/питаться легко” (*est’/pitat’sya legko*), a combination of the verb to eat with an ill-passing adverb. Some possible translations would be Russian equivalents “to eat light food” or “to eat little” that we see in the first and fourth translations.

The second example is taken from the phenomenon of *pseudogapping*. *Pseudogapping* is an ellipsis mechanism in which a part of the verb phrase is omitted. In the example at hand, the non-finite verb part “know” is omitted in the second conjunct of the construction. Instead, the auxiliary verb “do” is used as a substitute for the full verb. Verbal substitution is not common in Russian. Moreover, Russian does not employ auxiliary verbs (such as “to do” or “to be”) to form parallel elliptical constructions standard in English. The verb “does” in the second part of the sentence is translated as “сделает” (*sdelaet*) in the first incorrect Russian translation, leading to an impossible Russian phrasing. The second translation leaves out the subject “you” or “ты” (*ty*) in the conjunct resulting in a syntactically incorrect construction.

The last example contains a *resultative predicate*. *Resultatives* contain a verb with an adjective describing the result of the verb action. *Resultative predicates* usually require a significant construction change to get an equivalent translation in the target language. “He read the children to sleep” would be transformed in Russian as “on chital detyam pered snom” meaning “he read to the children before they were going to bed,” as in the third translation in the table or as “on chital detyam, chtobi oni spali” meaning “he read to the children so that they would sleep”.

5.4 Comparison with previous years

The progress of the systems’ accuracy for particular categories through the last years can be seen in Table 6 for German-English (since 2018) and Table 9 for English-German (since 2021). The calculation has been done based on the common test items without warnings over all these years, which is 4307 items for German-English and 3616 items for English-German. The general trend of this year suggests small but steady improvements for most systems and categories. In a few cases where the accuracies deteriorated, this is only for very few percentage points.

6 Conclusions and Outlook

This paper presents a fine-grained, linguistically motivated test suite to evaluate machine translation outputs. The test suite was applied to evaluate and compare the outputs of 29 machine translation systems in three different language pairs: German–English, English–German, and (for the first time) English–Russian. Altogether, almost 7,000 test sentences, structured in various linguistic categories and phenomena, were evaluated altogether across the three language pairs. Additionally, a comparison to the evaluation in previous years for the language pairs German–English and English–German was drawn.

The average accuracy for most categories and phenomena is relatively high for German–English and English–German, with only about 5 % room for improvement. As compared to last year, this is an improvement of around 5 %. For English–Russian, the average accuracy is not as high, yet still around 80 %.

The high average accuracies do not necessarily mean that the respective categories and phenomena no longer pose difficulties for MT. Instead, it could mean that the difficulty of the test sentences has become too easy over the past few years and should thus be increased. Therefore, we are currently constructing more complex test sentences for German–English and English–German. Further work also includes expanding the English–Russian test suite with more phenomena and more test sentences.

Acknowledgements

This research was supported by the Deutsche Forschungsgemeinschaft (DFG) through the project TextQ, by the German Federal Ministry of Education through the project SocialWear (grant num. 01IW20002) and by the project European Language Grid (ELG), which has received funding from the European Union's Horizon 2020 research and innovation programme (grant agreement no. 825627). We would like to thank Hans Uszkoreit, Aljoscha Burchardt, Ursula Strohriegel, Renlong Ai and He Wang for their prior contributions on the creation of the test suite.

References

Eleftherios Avramidis and Vivien Macketanz. 2022. Linguistically motivated evaluation of machine translation metrics based on a challenge set. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Eleftherios Avramidis, Vivien Macketanz, Arle Lommel, and Hans Uszkoreit. 2018. Fine-grained evaluation of Quality Estimation for Machine translation based on a linguistically motivated Test Suite. In *Proceedings of the AMTA 2018 Workshop on Translation Quality Estimation and Automatic Post-Editing*, pages 243–248, Boston, MA. Association for Machine Translation in the Americas.

Eleftherios Avramidis, Vivien Macketanz, Ursula Strohriegel, Aljoscha Burchardt, and Sebastian Möller. 2020. Fine-grained linguistic evaluation for state-of-the-art Machine Translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 346–356, Online. Association for Computational Linguistics.

Eleftherios Avramidis, Vivien Macketanz, Ursula Strohriegel, and Hans Uszkoreit. 2019. Linguistic Evaluation of German-English Machine Translation Using a Test Suite. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 445–454, Florence, Italy. Association for Computational Linguistics.

Ondřej Bojar, Jiří Mírovský, Kateřina Rysová, and Magdaléna Rysová. 2018. EvalD Reference-Less Discourse Evaluation for WMT18. In *Proceedings of the Third Conference on Machine Translation*, pages 545–549, Belgium, Brussels. Association for Computational Linguistics.

Aljoscha Burchardt, Vivien Macketanz, Jon Dehdari, Georg Heigold, Jan-Thorsten Peter, and Philip Williams. 2017. A Linguistic Evaluation of Rule-Based, Phrase-Based, and Neural MT Engines. *The Prague Bulletin of Mathematical Linguistics*, 108:159–170.

Franck Burlot, Yves Scherrer, Vinit Ravishankar, Ondřej Bojar, Stig-Arne Grönroos, Maarit Koponen, Tommi Nieminen, and François Yvon. 2018. The WMT'18 Morpheval test suites for English-Czech, English-German, English-Finnish and Turkish-English. In *Proceedings of the Third Conference on Machine Translation*, pages 550–564, Belgium, Brussels. Association for Computational Linguistics.

Silvie Cinkova and Ondřej Bojar. 2018. Testsuite on Czech–English Grammatical Contrasts. In *Proceedings of the Third Conference on Machine Translation*, pages 565–575, Belgium, Brussels. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondrej Bojar. 2021. Results of the WMT21 Metrics Shared Task: Evaluating Metrics with Expert-based Human Evaluations on TED and News Domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online.

Liane Guillou and Christian Hardmeier. 2016. PROTEST: A Test Suite for Evaluating Pronouns in Machine Translation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 636–643, Portorož, Slovenia. European Language Resources Association (ELRA).

Liane Guillou, Christian Hardmeier, Ekaterina Lapshinova-Koltunski, and Sharid Loáiciga. 2018. A Pronoun Test Suite Evaluation of the English–German MT Systems at WMT 2018. In *Proceedings of the Third Conference on Machine Translation*, pages 576–583, Belgium, Brussels. Association for Computational Linguistics.

Christian Hardmeier, Christine Basta, Marta R. Costa-jussà, Gabriel Stanovsky, and Hila Gonen, editors. 2022. *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*. Association for Computational Linguistics, United States.

Ulrich Heid and Elke Hildenbrand. 1991. Some practical experience with the use of test suites for the evaluation of SYSTRAN. In *the Proceedings of the Evaluators' Forum, Les Rasses*. Citeseer.

Pierre Isabelle, Colin Cherry, and George Foster. 2017. A Challenge Set Approach to Evaluating Machine Translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2496, Copenhagen, Denmark. Association for Computational Linguistics.

Margaret King and Kirsten Falkedal. 1990. Using test suites in evaluation of machine translation systems. In *Proceedings of the 13th conference on Computational Linguistics*, volume 2, pages 211–216, Morristown, NJ, USA. Association for Computational Linguistics.

Tom Kocmi, Tomasz Limisiewicz, and Gabriel Stanovsky. 2020. Gender coreference and bias evaluation at wmt 2020. In *Proceedings of the Fifth Conference on Machine Translation*, pages 357–364, Online. Association for Computational Linguistics.

Vivien Macketanz, Eleftherios Avramidis, Aljoscha Burchardt, and Hans Uszkoreit. 2018. Fine-grained evaluation of German-English Machine Translation based on a Test Suite. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 584–593, Belgium, Brussels. Association for Computational Linguistics.

Vivien Macketanz, Eleftherios Avramidis, Aljoscha Burchardt, He Wang, Renlong Ai, Shushen Manakhimova, Ursula Strohriegel, Sebastian Möller, and Hans Uszkoreit. 2022. A Linguistically Motivated Test Suite to Semi-Automatically Evaluate German–English Machine Translation Output. In *Proceedings of the Language Resources and Evaluation Conference*, pages 936–947, Marseille, France. European Language Resources Association.

Vivien Macketanz, Eleftherios Avramidis, Shushen Manakhimova, and Sebastian Möller. 2021. Linguistic Evaluation for the 2021 State-of-the-art Machine Translation Systems for German to English and English to German. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1059–1073, Online. Association for Computational Linguistics.

Maja Popović. 2019. Evaluating Conjunction Disambiguation on English-to-German and French-to-German WMT 2019 Translation Hypotheses. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 464–469, Florence, Italy. Association for Computational Linguistics.

Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. 2019. The MuCoW Test Suite at WMT 2019: Automatically Harvested Multilingual Contrastive Word Sense Disambiguation Test Sets for Machine Translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 470–480, Florence, Italy. Association for Computational Linguistics.

Annette Rios, Mathias Müller, and Rico Sennrich. 2018. The Word Sense Disambiguation Test Suite at WMT18. In *Proceedings of the Third Conference on Machine Translation*, pages 594–602, Belgium, Brussels. Association for Computational Linguistics.

Kateřina Rysová, Magdaléna Rysová, Tomáš Musil, Lucie Poláková, and Ondřej Bojar. 2019. A Test Suite and Manual Evaluation of Document-Level NMT at WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 455–463, Florence, Italy. Association for Computational Linguistics.

Yves Scherrer, Alessandro Raganato, and Jörg Tiedemann. 2020. The MUCOW word sense disambiguation test suite at WMT 2020. In *Proceedings of the Fifth Conference on Machine Translation*, pages 365–370, Online. Association for Computational Linguistics.

Tereza Vojtěchová, Michal Novák, Miloš Klouček, and Ondřej Bojar. 2019. SAO WMT19 Test Suite: Machine Translation of Audit Reports. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 481–493, Florence, Italy. Association for Computational Linguistics.

Andrew Way. 1991. Developer-Oriented Evaluation of MT Systems. In *Proceedings of the Evaluators' Forum*, pages 237–244, Les Rasses, Vaud, Switzerland. ISSCO.

Vilém Zouhar, Tereza Vojtěchová, and Ondřej Bojar. 2020. WMT20 Document-Level Markable Error Exploration. In *Proceedings of the Fifth Conference on Machine Translation*, pages 371–380, Online. Association for Computational Linguistics.

A German–English

category	count	Onl-B	Onl-W	LanBr	Onl-A	JDExp	Onl-Y	PROMT	Onl-G	LT22	avg
Ambiguity	81	**93.8**	84.0	**90.1**	84.0	**92.6**	84.0	77.8	**87.7**	43.2	81.9
Composition	49	93.9	**95.9**	93.9	**95.9**	**98.0**	**100.0**	**98.0**	**98.0**	65.3	93.2
Coordination & ellipsis	56	**92.9**	**94.6**	**91.1**	**94.6**	**94.6**	**94.6**	**92.9**	**94.6**	67.9	90.9
False friends	36	**77.8**	**80.6**	**83.3**	**83.3**	**66.7**	**77.8**	**80.6**	**83.3**	61.1	77.2
Function word	69	**91.3**	**89.9**	**89.9**	**88.4**	**92.8**	**84.1**	**89.9**	**91.3**	52.2	85.5
LDD & interrogatives	149	**89.3**	**86.6**	**89.3**	**88.6**	**91.9**	77.9	**88.6**	**90.6**	61.1	84.9
MWE	76	**81.6**	**89.5**	**78.9**	**82.9**	**81.6**	**82.9**	**78.9**	**80.3**	48.7	78.4
Named entitiy & terminology	87	**94.3**	**95.4**	**95.4**	**90.8**	**90.8**	**94.3**	**88.5**	**94.3**	78.2	91.3
Negation	19	**100.0**	**94.7**	**100.0**	**100.0**	**94.7**	**100.0**	**100.0**	**100.0**	68.4	95.3
Non-verbal agreement	60	**96.7**	**96.7**	**98.3**	**91.7**	**96.7**	**93.3**	88.3	88.3	61.7	90.2
Punctuation	59	**91.5**	**98.3**	89.8	**98.3**	89.8	**98.3**	**91.5**	66.1	67.8	87.9
Subordination	167	**93.4**	87.4	**92.8**	**88.0**	**92.2**	**92.8**	**90.4**	**91.6**	74.3	89.2
Verb tense/aspect/mood	4058	81.3	83.3	81.6	**84.9**	81.5	83.1	80.4	83.3	57.3	79.6
Verb valency	83	**84.3**	**83.1**	**81.9**	**83.1**	**88.0**	**81.9**	**81.9**	77.1	50.6	79.1
micro-average	5049	83.1	**84.6**	83.2	**85.7**	83.3	84.1	81.8	84.2	58.2	80.9
macro-average	5049	**90.1**	**90.0**	**89.7**	**89.6**	**89.4**	88.9	87.7	87.6	61.3	86.0

Table 5: Accuracies (%) of successful translations on the category level for German–English. Boldface indicates the significantly best performing systems per row.

		Onl-B					Onl-Y				PROMT			Onl-A					Onl-W		Onl-G				
category	count	2018	2019	2020	2021	2022	2018	2019	2021	2022	2019	2020	2022	2018	2019	2020	2021	2022	2021	2022	2018	2019	2020	2021	2022
Ambiguity	76	76.3	77.6	78.9	85.5	93.4	67.1	78.9	82.9	84.2	50.0	65.8	77.6	68.4	69.7	77.6	81.6	84.2	85.5	84.2	72.4	75.0	84.2	85.5	88.2
Composition	47	97.9	97.9	95.7	100.0	95.7	89.4	91.5	91.5	100.0	78.7	89.4	97.9	80.9	91.5	93.6	95.7	95.7	95.7	95.7	70.2	83.0	95.7	97.9	97.9
Coordination & ellipsis	33	87.9	87.9	90.9	90.9	93.9	87.9	87.9	90.9	90.9	81.8	87.9	87.9	87.9	87.9	87.9	90.9	90.9	90.9	90.9	51.5	66.7	75.8	90.9	90.9
False friends	36	75.0	77.8	80.6	75.0	77.8	66.7	91.7	75.0	77.8	72.2	72.2	80.6	72.2	72.2	69.4	83.3	83.3	86.1	80.6	72.2	72.2	77.8	80.6	83.3
Function word	61	78.7	78.7	91.8	88.5	93.4	90.2	90.2	83.6	85.2	85.2	91.8	90.2	83.6	88.5	86.9	90.2	90.2	93.4	90.2	50.8	91.8	91.8	93.4	91.8
LDD & interrogatives	73	83.6	83.6	89.0	94.5	91.8	83.6	79.5	90.4	84.9	74.0	83.6	89.0	76.7	75.3	82.2	89.0	89.0	90.4	91.8	64.4	72.6	90.4	89.0	90.4
MWE	65	72.3	72.3	76.9	76.9	81.5	69.2	70.8	73.8	81.5	56.9	69.2	76.9	64.6	66.2	70.8	80.0	81.5	87.7	87.7	64.6	67.7	78.5	78.5	78.5
Named entity & term.	58	91.4	91.4	87.9	91.4	96.6	91.4	89.7	93.1	94.8	84.5	91.4	94.8	89.7	89.7	94.8	94.8	94.8	96.6	98.3	89.7	87.9	91.4	94.8	96.6
Negation	16	93.8	93.8	100.0	100.0	100.0	100.0	100.0	100.0	100.0	93.8	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	62.5	100.0	100.0	100.0	100.0
Non-verbal agreement	55	87.3	87.3	87.3	98.2	96.4	80.0	81.8	85.5	92.7	70.9	81.8	89.1	78.2	83.6	83.6	92.7	92.7	96.4	96.4	58.2	81.8	90.9	90.9	89.1
Punctuation	35	97.1	97.1	94.3	100.0	94.3	100.0	100.0	100.0	100.0	85.7	97.1	94.3	100.0	100.0	100.0	100.0	100.0	97.1	97.1	82.9	82.9	85.7	85.7	85.7
Subordination	90	87.8	88.9	93.3	94.4	95.6	92.2	92.2	92.2	93.3	86.7	93.3	92.2	94.4	78.9	93.3	92.2	92.2	91.1	91.1	81.1	90.0	92.2	91.1	93.3
Verb tense/aspect/mood	3604	77.1	77.3	79.5	78.6	81.3	73.8	75.6	76.7	83.3	78.3	78.0	80.5	75.4	86.4	80.8	85.3	85.2	86.3	84.2	49.4	69.2	83.5	79.2	83.7
Verb valency	58	81.0	81.0	89.7	89.7	87.9	79.3	81.0	81.0	86.2	70.7	82.8	87.9	77.6	81.0	86.2	86.2	86.2	87.9	87.9	69.0	77.6	86.2	86.2	84.5
micro-average	4307	78.2	78.5	80.9	80.6	83.0	75.3	77.2	78.3	84.3	77.6	78.9	81.7	76.3	85.6	81.6	**86.0**	**86.0**	**87.2**	85.3	52.6	71.0	84.2	80.8	84.7
macro-average	4307	84.8	85.2	88.3	90.3	**91.4**	83.6	86.5	86.9	89.6	76.4	84.6	88.5	82.1	83.6	86.2	90.1	90.4	**91.8**	**91.1**	67.1	79.9	87.4	88.8	89.6

Table 6: Comparisons of the accuracy (%) of several German–English systems through the years.

categ	count	Onl-B	Onl-W	LanBr	Onl-A	JDExp	Onl-Y	PROMT	Onl-G	LT22	avg
Ambiguity	81	**93.8**	84.0	**90.1**	84.0	**92.6**	84.0	77.8	**87.7**	43.2	81.9
Lexical ambiguity	63	**95.2**	**88.9**	**92.1**	84.1	**90.5**	85.7	79.4	**88.9**	47.6	83.6
Structural ambiguity	18	**88.9**	66.7	83.3	83.3	**100.0**	77.8	72.2	83.3	27.8	75.9
Composition	49	93.9	**95.9**	93.9	**95.9**	**98.0**	**100.0**	**98.0**	**98.0**	65.3	93.2
Compound	29	**96.6**	**96.6**	**96.6**	**93.1**	**96.6**	**100.0**	**96.6**	**96.6**	75.9	94.3
Phrasal verb	20	**90.0**	**95.0**	**90.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	50.0	91.7
Coordination & ellipsis	56	**92.9**	**94.6**	**91.1**	**94.6**	**94.6**	**94.6**	**92.9**	**94.6**	67.9	90.9
Gapping	19	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**94.7**	**100.0**	68.4	95.9
Right node raising	19	**84.2**	**84.2**	**78.9**	**84.2**	**84.2**	**84.2**	**84.2**	**84.2**	42.1	78.9
Sluicing	18	94.4	100.0	94.4	100.0	100.0	100.0	100.0	100.0	94.4	98.1
False friends	36	**77.8**	**80.6**	**83.3**	**83.3**	**66.7**	**77.8**	**80.6**	**83.3**	61.1	77.2
Function word	69	**91.3**	**89.9**	**89.9**	**88.4**	**92.8**	**84.1**	**89.9**	**91.3**	52.2	85.5
Focus particle	24	**100.0**	**95.8**	**95.8**	**95.8**	**95.8**	**100.0**	87.5	**95.8**	70.8	93.1
Modal particle	25	**76.0**	**76.0**	**76.0**	**80.0**	**84.0**	**72.0**	**84.0**	**80.0**	56.0	76.0
Question tag	20	**100.0**	**100.0**	**100.0**	**90.0**	**100.0**	80.0	**100.0**	**100.0**	25.0	88.3
LDD & interrogatives	149	**89.3**	**86.6**	**89.3**	**88.6**	**91.9**	77.9	**88.6**	**90.6**	61.1	84.9
Extended adjective construction	14	**100.0**	**100.0**	**100.0**	**100.0**	**92.9**	**92.9**	**100.0**	**100.0**	64.3	94.4
Extraposition	17	64.7	64.7	64.7	64.7	76.5	70.6	64.7	52.9	52.9	64.1
Multiple connectors	19	84.2	94.7	84.2	89.5	78.9	78.9	84.2	89.5	94.7	86.5
Pied-piping	18	**88.9**	**83.3**	**88.9**	**88.9**	**88.9**	**88.9**	**88.9**	**88.9**	55.6	84.6
Polar question	18	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	72.2	**100.0**	**100.0**	83.3	95.1
Scrambling	17	**94.1**	**88.2**	**94.1**	**82.4**	**94.1**	**94.1**	**76.5**	**94.1**	29.4	83.0
Topicalization	17	82.4	52.9	82.4	82.4	**100.0**	**88.2**	**88.2**	**94.1**	58.8	81.0
Wh-movement	29	**96.6**	**100.0**	**96.6**	**96.6**	**100.0**	55.2	**100.0**	**100.0**	51.7	88.5
MWE	76	**81.6**	**89.5**	**78.9**	**82.9**	**81.6**	**82.9**	**78.9**	**80.3**	48.7	78.4
Collocation	19	**100.0**	**100.0**	**100.0**	**100.0**	**94.7**	**100.0**	**100.0**	**100.0**	52.6	94.2
Idiom	18	**38.9**	**61.1**	**22.2**	**27.8**	**33.3**	**27.8**	**16.7**	**27.8**	0.0	28.4
Prepositional MWE	20	**90.0**	**95.0**	**90.0**	**100.0**	**95.0**	**100.0**	**95.0**	**95.0**	65.0	91.7
Verbal MWE	19	**94.7**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**94.7**	73.7	95.9
Named entitiy & terminology	87	**94.3**	**95.4**	**95.4**	**90.8**	**90.8**	**94.3**	**88.5**	**94.3**	78.2	91.3
Date	19	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	94.7	99.4
Domainspecific term	20	**85.0**	**85.0**	**90.0**	**75.0**	**80.0**	**85.0**	**65.0**	**85.0**	50.0	77.8
Location	20	**95.0**	**100.0**	**95.0**	**95.0**	**95.0**	**95.0**	**95.0**	**95.0**	85.0	94.4
Measuring unit	19	**100.0**	**100.0**	**100.0**	**100.0**	**89.5**	**100.0**	**100.0**	**100.0**	84.2	97.1
Proper name	9	88.9	88.9	88.9	77.8	88.9	88.9	77.8	88.9	77.8	85.2
Negation	19	**100.0**	**94.7**	**100.0**	**100.0**	**94.7**	**100.0**	**100.0**	**100.0**	68.4	95.3
Non-verbal agreement	60	**96.7**	**96.7**	**98.3**	**91.7**	**96.7**	**93.3**	88.3	88.3	61.7	90.2
Coreference	19	**89.5**	**100.0**	**94.7**	84.2	**94.7**	84.2	78.9	78.9	57.9	84.8
External possessor	21	**100.0**	**95.2**	**100.0**	**90.5**	**95.2**	**95.2**	**90.5**	**90.5**	42.9	88.9
Internal possessor	20	**100.0**	**95.0**	**100.0**	**100.0**	**100.0**	**100.0**	**95.0**	**95.0**	85.0	96.7
Punctuation	59	**91.5**	**98.3**	89.8	**98.3**	89.8	**98.3**	**91.5**	66.1	67.8	87.9
Comma	20	100.0	95.0	95.0	100.0	100.0	100.0	100.0	100.0	100.0	98.9
Quotation marks	39	87.2	**100.0**	87.2	**97.4**	84.6	**97.4**	87.2	48.7	51.3	82.3

categ	count	Onl-B	Onl-W	LanBr	Onl-A	JDExp	Onl-Y	PROMT	Onl-G	LT22	avg
Subordination	167	**93.4**	87.4	**92.8**	**88.0**	**92.2**	**92.8**	**90.4**	**91.6**	74.3	89.2
Adverbial clause	20	**100.0**	85.0	**95.0**	**90.0**	**90.0**	**95.0**	85.0	**90.0**	85.0	90.6
Cleft sentence	20	**95.0**	**95.0**	**95.0**	**95.0**	**95.0**	**95.0**	**95.0**	**95.0**	60.0	91.1
Free relative clause	17	88.2	88.2	88.2	82.4	88.2	94.1	94.1	94.1	76.5	88.2
Indirect speech	15	**93.3**	73.3	**93.3**	**93.3**	**100.0**	**100.0**	**93.3**	**100.0**	60.0	89.6
Infinitive clause	19	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	84.2	98.2
Object clause	18	**100.0**	**100.0**	**94.4**	**94.4**	**94.4**	**100.0**	**100.0**	**100.0**	83.3	96.3
Pseudo-cleft sentence	20	85.0	65.0	85.0	70.0	85.0	70.0	75.0	75.0	60.0	74.4
Relative clause	18	83.3	94.4	83.3	77.8	83.3	83.3	83.3	77.8	88.9	84.0
Subject clause	20	**95.0**	85.0	**100.0**	**90.0**	**95.0**	**100.0**	**90.0**	**95.0**	70.0	91.1
Verb tense/aspect/mood	4058	81.3	83.3	81.6	**84.9**	81.5	83.1	80.4	83.3	57.3	79.6
Conditional	20	**95.0**	**90.0**	**95.0**	**95.0**	85.0	**90.0**	**100.0**	**95.0**	75.0	91.1
Ditransitive - future I	36	**100.0**	**94.4**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	86.1	97.8
Ditransitive - future I subjunctive II	36	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	86.1	98.5
Ditransitive - future II	36	**97.2**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	52.8	94.4
Ditransitive - future II subjunctive II	36	**100.0**	88.9	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**94.4**	98.1
Ditransitive - perfect	36	100.0	97.2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	99.7
Ditransitive - pluperfect	36	61.1	**97.2**	61.1	**88.9**	**97.2**	**86.1**	52.8	50.0	75.0	74.4
Ditransitive - pluperfect subjunctive II	36	**100.0**	**100.0**	**100.0**	**100.0**	83.3	**100.0**	**100.0**	**100.0**	69.4	94.8
Ditransitive - present	36	**100.0**	86.1	**100.0**	91.7	**94.4**	**94.4**	80.6	88.9	91.7	92.0
Ditransitive - preterite	36	**91.7**	**94.4**	**88.9**	75.0	**83.3**	**88.9**	80.6	80.6	77.8	84.6
Ditransitive - preterite subjunctive II	35	71.4	68.6	71.4	68.6	71.4	80.0	71.4	71.4	65.7	71.1
Imperative	19	**100.0**	**94.7**	**100.0**	**89.5**	**100.0**	**100.0**	**100.0**	**94.7**	63.2	93.6
Intransitive - future I	35	100.0	100.0	97.1	100.0	100.0	100.0	100.0	100.0	97.1	99.4
Intransitive - future I subjunctive II	36	100.0	97.2	100.0	100.0	100.0	97.2	100.0	100.0	97.2	99.1
Intransitive - future II	37	**100.0**	**100.0**	**100.0**	91.9	78.4	89.2	**100.0**	**97.3**	40.5	88.6
Intransitive - future II subjunctive II	35	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	40.0	93.3
Intransitive - perfect	83	**100.0**	90.4	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	60.2	94.5
Intransitive - pluperfect	35	**82.9**	**85.7**	**88.6**	**94.3**	57.1	**88.6**	57.1	77.1	37.1	74.3
Intransitive - pluperfect subjunctive II	29	**100.0**	**100.0**	**100.0**	**93.1**	**100.0**	**96.6**	**100.0**	**100.0**	37.9	92.0
Intransitive - present	35	**100.0**	65.7	**100.0**	**100.0**	**100.0**	**97.1**	**100.0**	**100.0**	71.4	92.7
Intransitive - preterite	66	**92.4**	**84.8**	**92.4**	**93.9**	**92.4**	**90.9**	**93.9**	**90.9**	65.2	88.6
Intransitive - preterite subjunctive II	36	**80.6**	63.9	**75.0**	**69.4**	66.7	**69.4**	**86.1**	**77.8**	36.1	69.4
Modal - future I	160	86.9	**88.8**	86.9	**93.1**	**89.4**	69.4	85.0	**89.4**	58.1	83.0
Modal - future I subjunctive II	151	**86.8**	**87.4**	**87.4**	**90.1**	82.8	**88.1**	80.8	**90.1**	49.0	82.5
Modal - perfect	165	73.9	70.9	73.9	80.0	**93.9**	79.4	82.4	75.2	3.6	70.4
Modal - pluperfect	146	2.7	**55.5**	**0.7**	37.0	14.4	34.2	13.0	25.3	1.4	20.5
Modal - pluperfect subjunctive II	144	**56.3**	**60.4**	**56.3**	**57.6**	41.7	**57.6**	**56.3**	**54.9**	31.9	52.5
Modal - present	173	**87.3**	**89.6**	**87.9**	**91.9**	**87.9**	83.2	79.8	**87.9**	79.2	86.1
Modal - preterite	169	**100.0**	92.9	**100.0**	**99.4**	98.2	97.6	95.3	97.6	89.9	96.8
Modal - preterite subjunctive II	146	80.8	78.8	80.1	82.2	82.2	74.0	79.5	76.7	73.3	78.6
Modal negated - future I	151	**95.4**	**95.4**	**94.7**	92.1	**94.7**	**96.7**	**93.4**	**97.4**	70.2	92.2
Modal negated - future I subjunctive II	171	83.0	85.4	81.9	84.2	83.6	91.8	79.5	**98.8**	51.5	82.2

categ	count	Onl-B	Onl-W	LanBr	Onl-A	JDExp	Onl-Y	PROMT	Onl-G	LT22	avg
Modal negated - perfect	165	77.0	84.8	73.3	78.8	**93.9**	76.4	82.4	70.9	14.5	72.5
Modal negated - pluperfect	131	13.7	**67.9**	14.5	24.4	8.4	28.2	4.6	26.0	0.0	20.9
Modal negated - pluperfect subjunctive II	153	**56.2**	**67.3**	**60.1**	**62.1**	41.2	**66.0**	**62.7**	**62.1**	32.7	56.7
Modal negated - present	157	89.8	92.4	**94.3**	**96.2**	**97.5**	87.9	84.7	87.3	88.5	90.9
Modal negated - preterite	169	**98.8**	94.7	**99.4**	**100.0**	97.6	**99.4**	94.1	97.0	79.3	95.6
Modal negated - preterite subjunctive II	141	**84.4**	**86.5**	**87.2**	**80.9**	**83.7**	**85.8**	**87.9**	**87.9**	75.2	84.4
Progressive	18	**94.4**	83.3	**94.4**	83.3	**94.4**	**100.0**	**88.9**	**88.9**	50.0	86.4
Reflexive - future I	35	**88.6**	80.0	**88.6**	**97.1**	**85.7**	**94.3**	**88.6**	**91.4**	57.1	85.7
Reflexive - future I subjunctive II	34	**85.3**	76.5	**85.3**	**97.1**	82.4	**94.1**	79.4	**85.3**	41.2	80.7
Reflexive - future II	36	75.0	66.7	**83.3**	**91.7**	58.3	**83.3**	**80.6**	**91.7**	41.7	74.7
Reflexive - future II subjunctive II	36	**83.3**	66.7	**86.1**	**88.9**	**80.6**	**88.9**	**72.2**	**86.1**	36.1	76.5
Reflexive - perfect	36	**91.7**	58.3	**86.1**	**97.2**	**94.4**	**91.7**	**91.7**	**94.4**	41.7	83.0
Reflexive - pluperfect	35	77.1	71.4	77.1	**94.3**	**82.9**	80.0	**88.6**	**85.7**	37.1	77.1
Reflexive - pluperfect subjunctive II	32	**84.4**	**75.0**	**87.5**	**90.6**	**78.1**	**87.5**	**78.1**	**84.4**	43.8	78.8
Reflexive - present	36	**97.2**	63.9	**94.4**	**97.2**	**88.9**	**94.4**	**86.1**	**91.7**	50.0	84.9
Reflexive - preterite	34	**91.2**	52.9	**91.2**	79.4	**88.2**	**82.4**	73.5	**94.1**	47.1	77.8
Reflexive - preterite subjunctive II	28	**100.0**	71.4	**100.0**	89.3	**96.4**	82.1	85.7	**100.0**	57.1	86.9
Transitive - future I	41	97.6	97.6	97.6	97.6	97.6	97.6	97.6	97.6	92.7	97.0
Transitive - future I subjunctive II	36	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	94.4	99.4
Transitive - future II	36	**100.0**	**97.2**	**100.0**	**100.0**	**97.2**	**97.2**	**100.0**	**100.0**	86.1	97.5
Transitive - future II subjunctive II	35	**100.0**	**97.1**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	85.7	98.1
Transitive - perfect	42	97.6	95.2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	99.2
Transitive - pluperfect	36	47.2	**100.0**	50.0	**94.4**	83.3	**94.4**	**94.4**	83.3	88.9	81.8
Transitive - pluperfect subjunctive II	36	**97.2**	**100.0**	**100.0**	**100.0**	86.1	**100.0**	**100.0**	**97.2**	80.6	95.7
Transitive - present	48	**100.0**	89.6	**100.0**	**100.0**	**100.0**	**100.0**	**97.9**	**100.0**	91.7	97.7
Transitive - preterite	36	**97.2**	86.1	**94.4**	**100.0**	**97.2**	83.3	88.9	**100.0**	80.6	92.0
Transitive - preterite subjunctive II	35	68.6	60.0	68.6	62.9	71.4	60.0	80.0	60.0	65.7	66.3
Verb valency	83	**84.3**	**83.1**	**81.9**	**83.1**	**88.0**	**81.9**	**81.9**	77.1	50.6	79.1
Case government	28	**92.9**	**92.9**	**92.9**	**92.9**	**85.7**	**89.3**	**92.9**	**89.3**	50.0	86.5
Mediopassive voice	20	**90.0**	**90.0**	**85.0**	**90.0**	**95.0**	**85.0**	**85.0**	65.0	50.0	81.7
Passive voice	19	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	73.7	97.1
Resultative predicates	16	43.8	37.5	37.5	37.5	68.8	43.8	37.5	43.8	25.0	41.7
micro-average	5049	83.1	**84.6**	83.2	**85.7**	83.3	84.1	81.8	84.2	58.2	80.9
phen. macro-average	5049	**88.4**	86.5	**88.2**	**88.9**	**88.1**	**88.1**	86.5	**88.1**	62.8	85.1
categ. macro-average	5049	**90.1**	**90.0**	**89.7**	**89.6**	**89.4**	88.9	87.7	87.6	61.3	86.0

Table 7: Accuracies (%) of successful translations on the phenomenon level for German–English. Boldface indicates the significantly best performing systems per row.

B English–German

category	count	JDExp	Onl-A	Onl-W	Onl-B	LanBr	Onl-G	PROMT	Onl-Y	OpenN	avg
Ambiguity	24	**91.7**	**87.5**	**95.8**	**91.7**	**83.3**	**83.3**	**79.2**	**79.2**	62.5	83.8
Coordination & ellipsis	74	78.4	79.7	67.6	**91.9**	**90.5**	**85.1**	**83.8**	79.7	66.2	80.3
False friends	38	92.1	84.2	89.5	86.8	84.2	89.5	84.2	84.2	78.9	86.0
Function word	42	97.6	97.6	100.0	97.6	97.6	97.6	97.6	97.6	95.2	97.6
MWE	110	**90.9**	84.5	**93.6**	**90.0**	83.6	85.5	80.0	82.7	80.9	85.8
Named entitiy & terminology	74	94.6	93.2	90.5	94.6	93.2	90.5	90.5	90.5	89.2	91.9
Negation	17	100.0	100.0	100.0	100.0	100.0	94.1	94.1	100.0	94.1	98.0
Non-verbal agreement	71	98.6	95.8	98.6	97.2	98.6	95.8	95.8	95.8	93.0	96.6
Punctuation	19	**100.0**	**94.7**	84.2	63.2	63.2	63.2	68.4	63.2	**100.0**	77.8
Subordination	162	**100.0**	**99.4**	98.1	**99.4**	**99.4**	**100.0**	**98.8**	93.2	98.1	98.5
Verb tense/aspect/mood	3009	97.4	98.1	96.2	**98.7**	**99.2**	97.6	**98.7**	95.2	83.6	96.1
Verb valency	83	**91.6**	**84.3**	**84.3**	**86.7**	**85.5**	**84.3**	**83.1**	**84.3**	77.1	84.6
micro-average	3723	96.7	96.7	95.2	**97.6**	**97.7**	96.3	96.9	93.8	84.1	95.0
macro-average	3723	**94.4**	91.6	91.5	91.5	89.9	88.9	87.9	87.1	84.9	89.7

Table 8: Accuracies (%) of successful translations on the category level for English–German. Boldface indicates the significantly best performing systems per row.

		Onl-A		Onl-W		Onl-B		Onl-G		Onl-Y	
category	count	2021	2022	2021	2022	2021	2022	2021	2022	2021	2022
Ambiguity	24	92	88	96	96	92	92	75	83	71	79
Coordination & ellipsis	75	69	80	65	64	84	91	73	84	68	75
False friends	39	85	85	87	90	82	87	82	90	85	85
Function word	41	98	98	100	100	100	98	98	98	98	98
MWE	109	83	85	93	95	89	90	79	86	80	83
Named entitiy & terminology	72	93	93	96	93	93	97	76	93	92	93
Negation	16	94	100	100	100	94	100	94	94	100	100
Non-verbal agreement	71	97	97	96	97	94	97	92	96	92	96
Punctuation	36	97	97	97	92	78	78	69	78	78	78
Subordination	161	99	99	98	98	99	99	95	99	94	93
Verb tense/aspect/mood	2885	96	98	97	96	99	99	95	98	92	95
Verb valency	87	83	86	89	85	86	89	75	86	79	86
micro-avg	3616	95	97	96	95	97	98	93	96	90	94
macro-avg	3616	90	92	93	92	91	93	84	90	86	88

Table 9: Comparisons of the accuracy (%) of several English–German systems through the years.

category/phenomenon	count	JDExp	Onl-A	Onl-W	Onl-B	LanBr	Onl-G	PROMT	Onl-Y	OpenN	avg
Ambiguity	24	**91.7**	**87.5**	**95.8**	**91.7**	**83.3**	**83.3**	**79.2**	**79.2**	62.5	83.8
Lexical ambiguity	24	**91.7**	**87.5**	**95.8**	**91.7**	**83.3**	**83.3**	**79.2**	**79.2**	62.5	83.8
Coordination & ellipsis	74	78.4	79.7	67.6	**91.9**	**90.5**	**85.1**	**83.8**	79.7	66.2	80.3
Gapping	14	78.6	85.7	64.3	92.9	92.9	78.6	85.7	71.4	64.3	79.4
Pseudogapping	6	83.3	50.0	66.7	83.3	83.3	83.3	66.7	50.0	50.0	68.5
Right node raising	11	90.9	100.0	90.9	90.9	90.9	90.9	100.0	81.8	90.9	91.9
Sluicing	14	**100.0**	78.6	**100.0**	**92.9**	78.6	78.6	78.6	78.6	78.6	84.9
Stripping	19	52.6	78.9	36.8	**94.7**	**100.0**	**89.5**	78.9	**89.5**	47.4	74.3
VP-ellipsis	10	80.0	70.0	60.0	90.0	90.0	90.0	90.0	90.0	70.0	81.1
False friends	38	92.1	84.2	89.5	86.8	84.2	89.5	84.2	84.2	78.9	86.0
Function word	42	97.6	97.6	100.0	97.6	97.6	97.6	97.6	97.6	95.2	97.6
Focus particle	23	95.7	95.7	100.0	95.7	95.7	95.7	95.7	95.7	91.3	95.7
Question tag	19	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
MWE	110	**90.9**	84.5	**93.6**	**90.0**	83.6	85.5	80.0	82.7	80.9	85.8
Collocation	17	100.0	100.0	100.0	94.1	88.2	100.0	94.1	94.1	88.2	95.4
Compound	19	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Idiom	18	**55.6**	**33.3**	**72.2**	**50.0**	**33.3**	22.2	**11.1**	22.2	16.7	35.2
Nominal MWE	18	94.4	88.9	94.4	100.0	94.4	100.0	88.9	88.9	94.4	93.8
Prepositional MWE	14	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Verbal MWE	24	95.8	87.5	95.8	95.8	87.5	91.7	87.5	91.7	87.5	91.2
Named entitiy & terminology	74	94.6	93.2	90.5	94.6	93.2	90.5	90.5	90.5	89.2	91.9
Date	16	100.0	100.0	100.0	100.0	100.0	100.0	100.0	93.8	100.0	99.3
Domainspecific term	9	77.8	88.9	77.8	77.8	77.8	77.8	77.8	77.8	88.9	80.2
Location	17	88.2	88.2	94.1	88.2	88.2	88.2	88.2	88.2	88.2	88.9
Measuring unit	18	**100.0**	**94.4**	83.3	**100.0**	**100.0**	**94.4**	**94.4**	**100.0**	**88.9**	95.1
Proper name	14	**100.0**	**92.9**	**92.9**	**100.0**	**92.9**	**85.7**	**85.7**	**85.7**	78.6	90.5
Negation	17	100.0	100.0	100.0	100.0	100.0	94.1	94.1	100.0	94.1	98.0
Non-verbal agreement	71	98.6	95.8	98.6	97.2	98.6	95.8	95.8	95.8	93.0	96.6
Coreference	25	96.0	88.0	96.0	92.0	96.0	96.0	88.0	88.0	88.0	92.0
Genitive	17	100.0	100.0	100.0	100.0	100.0	94.1	100.0	100.0	94.1	98.7
Possession	29	100.0	100.0	100.0	100.0	100.0	96.6	100.0	100.0	96.6	99.2
Punctuation	19	**100.0**	**94.7**	84.2	63.2	63.2	63.2	68.4	63.2	**100.0**	77.8
Quotation marks	19	**100.0**	**94.7**	84.2	63.2	63.2	63.2	68.4	63.2	**100.0**	77.8
Subordination	162	**100.0**	**99.4**	98.1	**99.4**	**99.4**	**100.0**	**98.8**	93.2	98.1	98.5
Adverbial clause	15	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	93.3	99.3
Cleft sentence	17	**100.0**	**94.1**	**100.0**	**94.1**	**100.0**	**100.0**	**94.1**	82.4	**94.1**	95.4
Contact clause	23	100.0	100.0	95.7	100.0	100.0	100.0	100.0	100.0	100.0	99.5
Indirect speech	13	100.0	100.0	84.6	100.0	100.0	100.0	100.0	100.0	100.0	98.3
Infinitive clause	18	100.0	100.0	100.0	100.0	100.0	100.0	100.0	94.4	94.4	98.8
Object clause	13	100.0	100.0	100.0	100.0	100.0	100.0	100.0	84.6	100.0	98.3
Pseudo-cleft sentence	16	**100.0**	**100.0**	**100.0**	**100.0**	**93.8**	**100.0**	**100.0**	75.0	**100.0**	96.5
Relative clause	34	100.0	100.0	100.0	100.0	100.0	100.0	97.1	97.1	100.0	99.3
Subject clause	13	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0

category/phenomenon	count	JDExp	Onl-A	Onl-W	Onl-B	LanBr	Onl-G	PROMT	Onl-Y	OpenN	avg
Verb tense/aspect/mood	3009	97.4	98.1	96.2	**98.7**	**99.2**	97.6	**98.7**	95.2	83.6	96.1
Conditional	17	94.1	94.1	94.1	88.2	88.2	94.1	94.1	94.1	94.1	92.8
Ditransitive - conditional I progressive	56	**100.0**	**100.0**	**96.4**	**100.0**	**100.0**	**100.0**	**98.2**	**100.0**	58.9	94.8
Ditransitive - conditional I simple	58	77.6	82.8	**98.3**	**100.0**	**100.0**	**98.3**	**96.6**	**100.0**	50.0	89.3
Ditransitive - conditional II progressive	56	100.0	100.0	100.0	100.0	100.0	100.0	100.0	96.4	96.4	99.2
Ditransitive - conditional II simple	57	**100.0**	**100.0**	**96.5**	**100.0**	**100.0**	**100.0**	**100.0**	94.7	94.7	98.4
Ditransitive - future I progressive	47	**97.9**	**97.9**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	89.4	98.3
Ditransitive - future I simple	99	94.9	**99.0**	**99.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	96.0	98.8
Ditransitive - future II progressive	51	94.1	**100.0**	92.2	**100.0**	**100.0**	92.2	**100.0**	58.8	37.3	86.1
Ditransitive - future II simple	57	84.2	94.7	**100.0**	**100.0**	**100.0**	93.0	**96.5**	77.2	36.8	86.9
Ditransitive - past perfect progressive	55	**96.4**	**100.0**	**98.2**	**100.0**	**100.0**	92.7	92.7	90.9	61.8	92.5
Ditransitive - past perfect simple	56	**98.2**	**100.0**	**96.4**	**100.0**	**100.0**	89.3	**98.2**	94.6	75.0	94.6
Ditransitive - past progressive	44	**100.0**	**100.0**	79.5	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**97.7**	97.5
Ditransitive - present perfect progressive	56	**100.0**	**100.0**	**98.2**	**100.0**	**100.0**	**100.0**	**100.0**	94.6	**96.4**	98.8
Ditransitive - present perfect simple	55	**100.0**	**100.0**	**98.2**	**100.0**	**100.0**	**100.0**	**100.0**	**96.4**	94.5	98.8
Ditransitive - present progressive	44	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	95.5	99.5
Ditransitive - simple past	73	**100.0**	**100.0**	**98.6**	**100.0**	**100.0**	**100.0**	**100.0**	**97.3**	95.9	99.1
Ditransitive - simple present	52	**100.0**	**100.0**	94.2	**100.0**	**100.0**	**98.1**	**100.0**	**100.0**	90.4	98.1
Gerund	21	100.0	100.0	95.2	100.0	100.0	100.0	100.0	100.0	95.2	98.9
Imperative	13	**100.0**	**100.0**	**92.3**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	61.5	94.9
Intransitive - conditional I progressive	27	**96.3**	**100.0**	85.2	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**92.6**	97.1
Intransitive - conditional I simple	29	**96.6**	**100.0**	**96.6**	**93.1**	**100.0**	**100.0**	**100.0**	**100.0**	89.7	97.3
Intransitive - conditional II progressive	22	**100.0**	**100.0**	81.8	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	98.0
Intransitive - conditional II simple	21	100.0	100.0	95.2	100.0	100.0	100.0	100.0	100.0	100.0	99.5
Intransitive - future I progressive	24	91.7	100.0	91.7	100.0	100.0	100.0	100.0	100.0	91.7	97.2
Intransitive - future I simple	64	95.3	**100.0**	89.1	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	93.8	97.6
Intransitive - future II progressive	24	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	20.8	62.5	87.0
Intransitive - future II simple	35	**97.1**	**100.0**	**100.0**	**100.0**	**100.0**	**97.1**	**97.1**	**94.3**	88.6	97.1
Intransitive - past perfect progressive	25	**92.0**	**100.0**	**96.0**	**100.0**	**100.0**	**96.0**	**96.0**	**100.0**	76.0	95.1
Intransitive - past perfect simple	33	**97.0**	**100.0**	**97.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	78.8	97.0
Intransitive - past progressive	28	100.0	100.0	100.0	92.9	100.0	100.0	100.0	96.4	100.0	98.8
Intransitive - present perfect progressive	2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Intransitive - present perfect simple	27	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Intransitive - present progressive	55	98.2	100.0	96.4	100.0	100.0	100.0	100.0	100.0	98.2	99.2
Intransitive - simple past	38	100.0	97.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	99.7
Intransitive - simple present	33	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Modal	271	**100.0**	**100.0**	98.9	98.5	**100.0**	**100.0**	**99.6**	98.5	96.3	99.1
Modal negated	270	**98.5**	**98.9**	97.4	**98.1**	**99.6**	**98.9**	**99.3**	**98.9**	95.2	98.3
Reflexive - conditional I progressive	34	**97.1**	**97.1**	91.2	**100.0**	**100.0**	91.2	**100.0**	**100.0**	58.8	92.8
Reflexive - conditional I simple	28	**92.9**	**92.9**	**92.9**	**100.0**	**100.0**	**100.0**	**100.0**	**92.9**	64.3	92.9
Reflexive - conditional II progressive	30	**100.0**	**100.0**	83.3	**96.7**	**96.7**	**100.0**	**100.0**	90.0	70.0	93.0
Reflexive - conditional II simple	31	**100.0**	**100.0**	87.1	**96.8**	**100.0**	**100.0**	**100.0**	90.3	90.3	96.1
Reflexive - future I progressive	32	**100.0**	**100.0**	**96.9**	**96.9**	**96.9**	**96.9**	**100.0**	**100.0**	75.0	95.8

category/phenomenon	count	JDExp	Onl-A	Onl-W	Onl-B	LanBr	Onl-G	PROMT	Onl-Y	OpenN	avg
Reflexive - future I simple	50	**100.0**	**100.0**	**96.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	90.0	98.4
Reflexive - future II progressive	27	**100.0**	**92.6**	88.9	**100.0**	**100.0**	**100.0**	**100.0**	48.1	55.6	87.2
Reflexive - future II simple	28	**100.0**	**96.4**	**92.9**	**100.0**	**100.0**	**100.0**	**100.0**	**96.4**	75.0	95.6
Reflexive - past perfect progressive	28	**100.0**	**92.9**	85.7	**92.9**	**96.4**	67.9	75.0	71.4	35.7	79.8
Reflexive - past perfect simple	28	**100.0**	**100.0**	**92.9**	**100.0**	**100.0**	85.7	**96.4**	82.1	53.6	90.1
Reflexive - past progressive	34	**100.0**	**100.0**	**100.0**	**97.1**	**97.1**	85.3	**100.0**	**97.1**	76.5	94.8
Reflexive - present perfect progressive	30	**100.0**	**100.0**	**100.0**	**96.7**	**100.0**	**100.0**	**100.0**	**100.0**	80.0	97.4
Reflexive - present perfect simple	31	100.0	100.0	93.5	100.0	100.0	100.0	100.0	100.0	96.8	98.9
Reflexive - present progressive	34	**94.1**	**94.1**	**97.1**	**94.1**	**97.1**	**85.3**	**91.2**	**97.1**	70.6	91.2
Reflexive - simple past	32	100.0	100.0	96.9	96.9	100.0	96.9	100.0	100.0	93.8	98.3
Reflexive - simple present	27	**100.0**	85.2	**100.0**	**100.0**	**96.3**	88.9	**100.0**	**96.3**	81.5	94.2
Transitive - future II progressive	29	**96.6**	**100.0**	**96.6**	**100.0**	**100.0**	**100.0**	**100.0**	37.9	44.8	86.2
Transitive - conditional I progressive	29	**100.0**	**100.0**	89.7	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	69.0	95.4
Transitive - conditional I simple	30	**93.3**	83.3	**100.0**	**100.0**	**100.0**	**100.0**	**93.3**	**100.0**	76.7	94.1
Transitive - conditional II progressive	29	**100.0**	**100.0**	**96.6**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	79.3	97.3
Transitive - conditional II simple	28	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	92.9	99.2
Transitive - future I progressive	26	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Transitive - future I simple	50	100.0	98.0	100.0	100.0	100.0	98.0	98.0	98.0	100.0	99.1
Transitive - future II simple	32	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**96.9**	46.9	93.8
Transitive - past perfect progressive	28	**100.0**	**100.0**	71.4	**100.0**	**100.0**	**96.4**	**100.0**	**100.0**	53.6	91.3
Transitive - past perfect simple	28	**100.0**	**100.0**	**96.4**	**100.0**	**100.0**	**96.4**	**100.0**	**100.0**	67.9	95.6
Transitive - past progressive	28	39.3	39.3	**92.9**	57.1	42.9	60.7	71.4	**96.4**	57.1	61.9
Transitive - present perfect progressive	30	**100.0**	**100.0**	**96.7**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	83.3	97.8
Transitive - present perfect simple	35	**100.0**	**100.0**	**97.1**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	85.7	98.1
Transitive - present progressive	35	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	88.6	98.7
Transitive - simple past	38	**97.4**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	92.1	98.8
Transitive - simple present	35	100.0	100.0	100.0	97.1	100.0	100.0	100.0	100.0	100.0	99.7
Verb valency	83	**91.6**	**84.3**	**84.3**	**86.7**	**85.5**	**84.3**	**83.1**	**84.3**	77.1	84.6
Case government	18	94.4	94.4	88.9	94.4	94.4	94.4	94.4	88.9	94.4	93.2
Catenative verb	18	100.0	100.0	94.4	100.0	100.0	100.0	100.0	100.0	88.9	98.1
Middle voice	15	73.3	53.3	73.3	60.0	60.0	53.3	46.7	53.3	26.7	55.6
Passive voice	15	100.0	100.0	93.3	100.0	100.0	100.0	100.0	100.0	100.0	99.3
Resultative	17	88.2	70.6	70.6	76.5	70.6	70.6	70.6	76.5	70.6	73.9
micro-average	3723	96.7	96.7	95.2	**97.6**	**97.7**	96.3	96.9	93.8	84.1	95.0
phen. macro-average	3723	**95.8**	94.8	93.4	**96.3**	**95.9**	94.5	94.8	91.4	82.4	93.3
categ. macro-average	3723	**94.4**	91.6	91.5	91.5	89.9	88.9	87.9	87.1	84.9	89.7

Table 10: Accuracies (%) of successful translations on the phenomenon level for English–German. Boldface indicates the significantly best performing systems per row.

C English–Russian

category	count	Onl-W	Onl-G	Onl-B	JDExp	LanBr	Huawe	Onl-A	PROMT	Onl-Y	SRPOL	eTran	avg
Ambiguity	11	**100.0**	**90.9**	**90.9**	**90.9**	**81.8**	**81.8**	72.7	63.6	**81.8**	72.7	63.6	81.0
Coordination & ellipsis	27	**70.4**	**77.8**	**55.6**	**74.1**	48.1	**55.6**	44.4	**59.3**	**51.9**	**55.6**	**51.9**	58.6
False friends	5	80.0	80.0	80.0	80.0	60.0	60.0	60.0	60.0	60.0	60.0	60.0	67.3
Function word	10	**100.0**	**90.0**	**90.0**	70.0	**80.0**	**80.0**	**80.0**	**80.0**	**90.0**	**80.0**	**80.0**	83.6
MWE	39	76.9	74.4	76.9	74.4	66.7	64.1	66.7	64.1	66.7	59.0	61.5	68.3
Named entitiy & terminology	26	**73.1**	**88.5**	**84.6**	**84.6**	**84.6**	65.4	**69.2**	**73.1**	**73.1**	65.4	65.4	75.2
Negation	5	100.0	80.0	100.0	80.0	100.0	80.0	80.0	80.0	80.0	100.0	80.0	87.3
Non-verbal agreement	23	73.9	78.3	73.9	82.6	69.6	73.9	69.6	73.9	69.6	69.6	65.2	72.7
Punctuation	5	100.0	100.0	100.0	100.0	80.0	100.0	100.0	100.0	80.0	80.0	80.0	92.7
Subordination	49	**91.8**	**89.8**	**91.8**	**93.9**	**89.8**	79.6	81.6	81.6	79.6	81.6	75.5	85.2
Verb tense/aspect/mood	68	70.6	72.1	72.1	76.5	75.0	67.6	73.5	73.5	66.2	70.6	67.6	71.4
Verb valency	32	87.5	87.5	78.1	84.4	75.0	75.0	84.4	71.9	78.1	75.0	68.8	78.7
micro-average	300	**80.3**	**81.3**	**78.7**	**81.7**	75.0	70.7	72.3	72.3	71.0	70.3	67.0	74.6
macro-average	300	**85.4**	**84.1**	**82.8**	**82.6**	75.9	73.6	73.5	73.4	73.1	72.5	68.3	76.8

Table 11: Accuracies (%) of successful translations on the category level for English–Russian. Boldface indicates the significantly best performing systems per row.

category/phenomenon	count	Onl-W	Onl-G	Onl-B	JDExp	LanBr	Huawe	Onl-A	PROMT	Onl-Y	SRPOL	eTran	avg
Ambiguity	11	**100.0**	**90.9**	**90.9**	**90.9**	**81.8**	**81.8**	72.7	63.6	**81.8**	72.7	63.6	81.0
Lexical ambiguity	11	**100.0**	**90.9**	**90.9**	**90.9**	**81.8**	**81.8**	72.7	63.6	**81.8**	72.7	63.6	81.0
Coordination & ellipsis	27	**70.4**	**77.8**	**55.6**	**74.1**	48.1	**55.6**	44.4	**59.3**	**51.9**	**55.6**	**51.9**	58.6
Gapping	5	**40.0**	**80.0**	**20.0**	**80.0**	**20.0**	**60.0**	0.0	**60.0**	**40.0**	**60.0**	**40.0**	45.5
Pseudogapping	6	**50.0**	**83.3**	**50.0**	**66.7**	**16.7**	0.0	0.0	**16.7**	**16.7**	0.0	**16.7**	28.8
Right node raising	5	100.0	80.0	80.0	100.0	80.0	100.0	80.0	80.0	80.0	80.0	80.0	85.5
Sluicing	3	**100.0**	**66.7**	**66.7**	**33.3**	**66.7**	**33.3**	**66.7**	**66.7**	**66.7**	**100.0**	0.0	60.6
Stripping	5	80.0	80.0	40.0	60.0	40.0	60.0	60.0	80.0	60.0	80.0	80.0	65.5
VP-ellipsis	3	66.7	66.7	100.0	100.0	100.0	100.0	100.0	66.7	66.7	33.3	100.0	81.8
False friends	5	80.0	80.0	80.0	80.0	60.0	60.0	60.0	60.0	60.0	60.0	60.0	67.3
Function word	10	**100.0**	**90.0**	**90.0**	70.0	**80.0**	**80.0**	**80.0**	**80.0**	**90.0**	**80.0**	**80.0**	83.6
Focus particle	5	100.0	80.0	80.0	80.0	80.0	100.0	100.0	80.0	100.0	100.0	100.0	90.9
Question tag	5	100.0	100.0	100.0	60.0	80.0	60.0	60.0	80.0	80.0	60.0	60.0	76.4
MWE	39	76.9	74.4	76.9	74.4	66.7	64.1	66.7	64.1	66.7	59.0	61.5	68.3
Collocation	8	75.0	62.5	62.5	87.5	62.5	50.0	50.0	62.5	62.5	62.5	37.5	61.4
Compound Adjectives	6	100.0	100.0	100.0	83.3	100.0	100.0	100.0	66.7	83.3	66.7	100.0	90.9
Idiom	8	25.0	50.0	50.0	37.5	25.0	25.0	37.5	37.5	25.0	12.5	12.5	30.7
Nominal MWE	6	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Prepositional MWE	5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Verbal MWE	6	83.3	50.0	66.7	50.0	33.3	33.3	33.3	33.3	50.0	33.3	50.0	47.0
Named entitiy & terminology	26	**73.1**	**88.5**	**84.6**	**84.6**	**84.6**	65.4	**69.2**	**73.1**	**73.1**	65.4	65.4	75.2
Date	5	100.0	100.0	100.0	100.0	100.0	80.0	80.0	80.0	100.0	100.0	100.0	94.5

category/phenomenon	count	Onl-W	Onl-G	Onl-B	JDExp	LanBr	Huawe	Onl-A	PROMT	Onl-Y	SRPOL	eTran	avg
Domainspecific Term	5	80.0	100.0	100.0	80.0	100.0	40.0	60.0	60.0	40.0	40.0	40.0	67.3
Location	5	40.0	60.0	80.0	80.0	80.0	60.0	80.0	60.0	80.0	60.0	60.0	67.3
Measuring unit	5	80.0	80.0	80.0	80.0	80.0	80.0	60.0	100.0	80.0	60.0	60.0	76.4
Proper name	6	66.7	100.0	66.7	83.3	66.7	66.7	66.7	66.7	66.7	66.7	66.7	71.2
Negation	5	100.0	80.0	100.0	80.0	100.0	80.0	80.0	80.0	80.0	100.0	80.0	87.3
Non-verbal agreement	23	73.9	78.3	73.9	82.6	69.6	73.9	69.6	73.9	69.6	69.6	65.2	72.7
Anaphora agreement	7	57.1	71.4	42.9	71.4	42.9	42.9	42.9	57.1	42.9	42.9	42.9	50.6
Coreference	5	80.0	80.0	100.0	100.0	80.0	100.0	80.0	80.0	60.0	100.0	80.0	85.5
Genitive	6	83.3	83.3	83.3	83.3	83.3	83.3	83.3	83.3	83.3	66.7	66.7	80.3
Possession	5	80.0	80.0	80.0	80.0	80.0	80.0	80.0	80.0	100.0	80.0	80.0	81.8
Punctuation	5	100.0	100.0	100.0	100.0	80.0	100.0	100.0	100.0	80.0	80.0	80.0	92.7
Direct Speech	5	100.0	100.0	100.0	100.0	80.0	100.0	100.0	100.0	80.0	80.0	80.0	92.7
Subordination	49	**91.8**	**89.8**	**91.8**	**93.9**	**89.8**	79.6	81.6	81.6	79.6	81.6	75.5	85.2
Adverbial clause	5	80.0	80.0	80.0	80.0	80.0	80.0	80.0	80.0	40.0	60.0	80.0	74.5
Cleft sentence	5	80.0	80.0	80.0	80.0	60.0	40.0	40.0	80.0	60.0	60.0	40.0	63.6
Contact clause	5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Indirect speech	4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Infinitive clause	5	100.0	80.0	80.0	80.0	80.0	80.0	80.0	80.0	80.0	60.0	80.0	80.0
Object clause	5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	80.0	80.0	100.0	100.0	96.4
Participle clause	5	100.0	100.0	100.0	100.0	100.0	80.0	100.0	100.0	60.0	100.0	100.0	94.5
Pseudo-cleft sentence	5	80.0	80.0	100.0	100.0	100.0	60.0	80.0	80.0	100.0	100.0	20.0	81.8
Relative clause	5	80.0	80.0	80.0	100.0	80.0	80.0	40.0	60.0	80.0	60.0	60.0	72.7
Subject clause	5	100.0	100.0	100.0	100.0	100.0	80.0	100.0	60.0	100.0	80.0	80.0	90.9
Verb tense/aspect/mood	68	70.6	72.1	72.1	76.5	75.0	67.6	73.5	73.5	66.2	70.6	67.6	71.4
Conditional	5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	80.0	98.2
Ditransitive	16	75.0	81.3	81.3	93.8	93.8	87.5	87.5	87.5	68.8	81.3	81.3	83.5
Gerund	5	80.0	80.0	100.0	80.0	80.0	80.0	100.0	80.0	100.0	80.0	60.0	83.6
Imperative	5	80.0	100.0	60.0	60.0	60.0	60.0	60.0	60.0	60.0	60.0	60.0	65.5
Intransitive	16	43.8	43.8	43.8	50.0	50.0	43.8	43.8	56.3	50.0	43.8	43.8	46.6
Reflexive	5	100.0	100.0	80.0	100.0	80.0	80.0	80.0	80.0	80.0	80.0	80.0	85.5
Transitive	16	68.8	62.5	75.0	75.0	75.0	56.3	75.0	68.8	56.3	75.0	75.0	69.3
Verb valency	32	87.5	87.5	78.1	84.4	75.0	75.0	84.4	71.9	78.1	75.0	68.8	78.7
Case government	5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	80.0	80.0	96.4
Catenative verb	7	85.7	85.7	71.4	71.4	57.1	71.4	71.4	57.1	71.4	71.4	57.1	70.1
Impersonal Subject	5	100.0	100.0	100.0	100.0	100.0	80.0	100.0	100.0	100.0	100.0	100.0	98.2
Middle voice	5	40.0	60.0	40.0	40.0	40.0	40.0	60.0	40.0	40.0	40.0	40.0	43.6
Passive voice	5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Resultative	5	100.0	80.0	60.0	100.0	60.0	60.0	80.0	40.0	60.0	60.0	40.0	67.3
micro-average	300	**80.3**	**81.3**	**78.7**	**81.7**	75.0	70.7	72.3	72.3	71.0	70.3	67.0	74.6
phen. macro-average	300	**83.2**	**84.0**	**81.0**	**83.2**	76.7	72.7	74.2	73.8	73.4	72.2	68.3	76.6
categ. macro-average	300	**85.4**	**84.1**	**82.8**	**82.6**	75.9	73.6	73.5	73.4	73.1	72.5	68.3	76.8

Table 12: Accuracies (%) of successful translations on the phenomenon level for English–Russian. Boldface indicates the significantly best performing systems per row.

Automated Evaluation Metric for Terminology Consistency in MT

Kirill Semenov and **Ondřej Bojar**
Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské námestí 25, 118 00 Prague, Czech Republic
kir.semenow@yandex.ru
bojar@ufal.mff.cuni.cz

Abstract

The most widely used metrics for machine translation tackle sentence-level evaluation. However, at least for professional domains such as legal texts, it is crucial to measure the consistency of translation of terms throughout the whole text.

This paper introduces an automated metric for term consistency evaluation in machine translation (MT). To demonstrate the metric's performance, we used the Czech-to-English translated texts from the ELITR 2021 agreement corpus and the outputs of the MT systems that took part in WMT21 and WMT22 News Tasks. We show different modes of our evaluation algorithm and try to interpret the differences in the ranking of the translation systems based on standard sentence-level metrics and our approach. We also demonstrate that the proposed metric scores significantly differ from the widespread automated metric scores, and correlate with human assessment.

1 Introduction

Throughout the last decade, the quality of machine translation (MT) has improved significantly, and it is becoming a common phenomenon for various neural MT (NMT) systems to get better scores in manual direct assessment and other metrics than reference human translations (Akhbardeh et al., 2021; Bojar et al., 2018). However, such figures are obtained when the MT outputs are evaluated on the sentence level (i.e., each sentence is assessed separately, without context); in document-level evaluation, human translations typically remain the best, although exceptions exist (Popel et al., 2020). We can explain this situation by the fact that most of the current state-of-the-art NMT systems translate documents sentence by sentence, which thus can provoke inconsistencies in the translation of different linguistic elements – from anaphoric pronouns to named entities and terminology. We focus on the latter.

While term inconsistencies can be tolerable for the general spheres of communication, they are unacceptable for several professional domains, especially legal texts, where the coherent usage of terms is the ultimate characteristic.

In the case of the term translation in the legal domain, the goal of the MT system can be split into several parts:

1. To translate one source term to only one target term (we will call this property "consistency");
2. To ensure that every source term is mapped to a distinct target term (we will call this property "unambiguity");
3. To ensure that the target term is an adequate translation of the source term in general.

In this paper, we present a novel metric that focuses on the consistency and unambiguity of terms, whereas measuring the third parameter, adequacy, is delegated to the mainstream automated metrics such as BLEU (Papineni et al., 2002) and chrF (Popović, 2015). Our proposed metric can be applied automatically, and it needs a small amount of human preprocessing and annotation (for instance, it does not require reference translation of the sentences). However, it can include manually tuned parameters as a variable.

In Section 2, we describe the background in the field of the term consistency in MT; in Section 3 we introduce the algorithm of our metric; in Section 4 we present the data on which the metric is applied; in Section 5 we discuss the results and compare them to the widespread automated metrics in MT. Limitations of our method are highlighted in Section 6.

2 Background

Scholars have been drawing attention to document-level consistency for over a decade. For instance,

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 450 - 457
December 7-8, 2022 ©2022 Association for Computational Linguistics

Hardmeier (2012) presents a number of discourse-related phenomena (such as pronoun use and verb tense modeling) that should be taken into account, as well as an overview of the metrics that were designed to catch the consistency (by that moment, they did not correlate with human judgments much). Since this time, there have been various experiments in enhancing the sentence-level MT models for better consistency, by a variety of means, from hierarchical approaches (Ture et al., 2012) to post-editing the output sentences (Voita et al., 2019b). Notably, the main focus of the proposed systems tends to be on the discourse-related features of texts, such as verb forms, anaphora, ellipsis, named entities, etc. (Voita et al., 2019a), rather than on the terminology consistency.

There has also been progress in designing the evaluation for lexical consistency in domain-specific spheres. For example, the creators of SAO WMT test suite (Vojtěchová et al., 2019) point out that the most accurate evaluation for the audit reports is performed manually by professionals in the field, while neither the automated metrics nor the direct evaluation by non-experts gives valuable information about the ranking of the systems' quality. The same authors in 2020 introduced the concept of the "markables": the linguistic elements to which the human annotators have to pay special attention (Zouhar et al., 2020). In the paper, they considered the domains of Sublease, News, and Audit, and the main markables were the crucial terms in the document. The research reaffirms that the automated metrics such as BLEU are not very informative with respect to term consistency, while the non-professional annotators cannot spot the domain-specific inconsistencies; however, the additional annotation of the "markables" allows even the "lay" annotators to keep in mind the necessary terms, which makes the manual annotation more accurate and informative.

Another notable research by Alam et al. (2021) presents the ideas for automated metrics for the term consistency evaluation, namely, exact match accuracy, window overlap, and TER with bigger penalties for terms. The results of this approach, tested on the domain of medical texts about COVID-19, show a correlation with human professional judgments; however, for most of the metrics, reference translations or at least term dictionaries are necessary. Thus, the relevance of designing more automated metrics in the field is still valid.

3 Metric

Before explaining the metric in detail, we will reiterate our aim. Our first objective is to reward translation consistency (i.e., penalize the one-to-many correspondences in source-to-target term pairs). Secondly, we want translated terms to be unambiguous (i.e., we should penalize the many-to-one correspondences in source-to-target term pairs). Optionally, we also want to include adequacy in our estimation (i.e., penalize the inappropriate translation of the term); otherwise, we will rely on the widely accepted metrics for adequacy. Finally, we want the algorithm to be as automatic as possible, i.e., to avoid the necessity of human annotation on any level. To meet these demands, we introduce the following pipeline.

1. **General preprocessing**: We tokenize the texts. The tokenization needs to be consistent in both source and target texts to run the alignment algorithms (Step 3 below).

2. **Source terms extraction**: We extract "crucial" terms in the source text. The task can be reduced to keyword extraction, which has various approaches. In our study, we used the manual method based on regular expressions: in legal-like texts, the terms relevant for the document are announced uniformly at the beginning of the document (for example, by the phrases "hereinafter referred as..."). We justify this choice in Section 6. As a result of this step, we get a set of the terms that occur in the text (hereinafter: src term set), and, for each sentence, we get a list of terms that appear there.

3. **Term Alignment**: For automation, we suggest using any word alignment algorithm. In this experiment, we used fast-align algorithm introduced by Dyer et al. (2013). Now, for each text separately, we extract the alignments of the source terms obtained in Step 1.[1] At the end of this step, for each document, we have, firstly, lists of aligned target terms in each sentence, secondly, the dictionary of source terms and the counts of their corresponding alignments in this text (hereinafter: src-tgt dict).

[1] To create a better word alignment, we firstly collect all outputs of the same system into one text, apply fast-align to such big texts, and then split the alignments back to the initial document level.

4. **Choosing the “pseudo-reference” translations**: To measure the performance of the MT system, we have to compare the real occurrences of the translations (obtained in Step 3, hereinafter called “candidate” translations) to the translations that we expect to be used throughout the text (we call them “pseudo-reference” translations). Choosing the pseudo-reference translation is the trickiest element of the task. However, we can introduce several solutions to it. On the one hand, we can count the first occurrence of each translated term as the pseudo-reference. This is reasonable in the logic of legal texts, where the terms are “introduced” at the beginning and consistently used afterwards. On the other hand, we can choose the most frequent translation of the term to be the correct translation. In our experiment, we tried both approaches, which are easily done by the src-tgt dict or by the lists of the target terms for each sentence in the text. As a result of this step, we obtain the list of the “pseudo-reference” target terms for each sentence. Notably, the choice of the “pseudo-reference” terminology is calculated separately for each document.
5. **Evaluation**: After the four steps, the final data structure consists of quintuples, where each quintuple consists of the source sentence, the target sentence, and three lists: of the source terms, of the “candidate” occurrences of the translated source terms, and of the “pseudo-reference” translations. We can represent them as a variant of the TORT annotation (term-only reference translation, introduced by Bafna et al., 2021), where for each MT output sentence, there is a list of crucial reference terms instead of the whole text. Such lists of lists of “candidate” and “pseudo-reference” occurrences can be measured by the widespread data science metrics – multi-class precision, recall, true positive rate, etc. For better granularity, we also suggest grouping the lists by the source terms and counting the percentage of the correct occurrences of the exact term (hereinafter we call it "our" or "our own" metric).

Therefore, the main novelty of our approach is not the metric itself but an algorithm for automatizing the data collection for applying the widespread metrics.

4 Data

We used the data from the ELITR agreement test suite to test the metric. The test suite consists of various short agreement documents, namely, 18 purchase agreements, 13 lease and sublease agreements, and two agreements on renting or using the software. All documents have Czech as the source language and English as the target language; only for three files, the reference English translations are provided. As the MT outputs, we used the results of seven MT systems that took part in 2021 and 2022 competitions on this test suite. Detailed information about the systems is presented in Akhbardeh et al. (2021) and **?**, and the test suite texts are available online.[2]

5 Results and Discussion

In this section, we firstly comment on the absolute scores of the different variants of the proposed metric; secondly, we compare the ranking of the MT systems by our metric and by the ones represented in the findings of WMT21 and WMT22.

5.1 Proposed Metric Scores

Speaking about the absolute scores (see Table 1), we can see that for both years, if we fix formula that we use (either F1 or our own metric), the most frequent pseudo-reference initialization is regularly higher than the first-occurrence one (1-3% for F1; 3-5% for our metric). If we fix the pseudo-reference initialization and compare different formulas, the difference is bigger and varies between 7-9%. This can be a reflection of the fact that the NMT models are sentence based. The reason is following: if a model has a pre-trained distribution of translations for each term, then it may tend to choose the same likeliest translation for the term in the majority of the sentences. Thus, such likeliest translations will be most frequent in the src-tgt dicts, and will be chosen as "pseudo-references" in case of the most frequent initialization.

If we take into account the ranking of the algorithms, we can see that the big difference tends to be between the variants with different pseudo-reference choice. Kendall’s tau paired comparisons between the variants support this hypothesis: the most correlating rankings are the F1 and our metric with first-occurrence initialization, next best correlation is between the F1 and our metric with the

[2] https://github.com/ELITR/agreement-corpus

Year	MT System	1st; F1	1st; Own	Freq; F1	Freq; Own	1st; F1 rank	1st; Own rank	Freq; F1 rank	Freq; Own rank
2021	**CUNI-Doc Transformer**	0.897	0.804	0.915	0.835	3	4	4	4
	CUNI-Transformer2018	0.857	0.776	0.895	0.827	8	7	8	7
	Facebook-AI	**0.907**	**0.838**	**0.930**	**0.871**	1	1	1	1
	Online-A	0.883	0.795	0.914	0.829	4	5	5	6
	Online-B	0.880	0.792	0.925	0.852	6	6	2	2
	Online-G	0.871	0.771	0.900	0.811	7	8	6	8
	Online-W	0.881	0.807	0.898	0.831	5	3	7	5
	Online-Y	0.900	0.813	0.921	0.840	2	2	3	3
2022	**ALMAnaCH-Inria**	0.816	0.688	0.885	0.807	11	11	10	9
	CUNI-Doc Transformer	0.897	0.805	0.916	0.836	4	6	4	6
	CUNI-Transformer	0.848	0.751	0.882	0.790	10	10	11	11
	JDExplore Academy	0.899	0.817	**0.928**	**0.863**	3	4	1	1
	Lan-Bridge	**0.902**	0.826	0.918	0.846	2	2	3	2
	Online-A	0.877	0.773	0.924	0.836	7	7	2	7
	Online-B	**0.902**	**0.831**	0.912	0.842	1	1	5	4
	Online-G	0.871	0.772	0.898	0.807	8	8	8	10
	Online-W	0.889	0.816	0.903	0.838	6	5	7	5
	Online-Y	0.860	0.767	0.892	0.809	9	9	9	8
	SHOPLINE-PL	0.895	0.822	0.910	0.845	5	3	6	3

Table 1: Scores of different metric variants. The first position in the column name denotes the method for choice of pseudo-reference ("Freq" for "most frequent translation", "1st" for "first occurrence"); the second means the metric ("F1" for F1 score and "Own" for our own metric – averaged percentage of the correct hits per term). The last four columns show the ranking of the systems.

Compared Setups	τ **2021**	τ **2022**
1st;F1 VS 1st;Own	.786*	.891*
1st;F1 VS Freq;F1	.643*	.636*
1st;F1 VS Freq;Own	.571	.673*
1st;Own VS Freq;F1	.429	.527*
1st;Own VS Freq;Own	.643*	.709*
Freq;F1 VS Freq;Own	.786*	.600*

Table 2: Pairwise Kendall's Tau correlations between the rankings of the scores obtained by different variants of our algorithm. The first column shows the pairs of variants we compare (separated by "VS"). The second and the third columns show Kendall's Tau scores; the asterisk denotes the values that are statistically significant for the null hypothesis of $\tau = 0 (p < 0.05)$.

Metrics Compared	τ **2021**	τ **2022**
1st;F1 VS BLEU	.357	-.527
1st;F1 VS chrf	.286	
1st;F1 VS DA	.714*	N/A
1st;Own VS BLEU	.143	-.636
1st;Own VS chrf	.071	
1st;Own VS DA	.500	N/A
Freq;F1 VS BLEU	.143	-.527
Freq;F1 VS chrf	.071	
Freq;F1 VS DA	.786*	N/A
Freq;Own VS BLEU	-.071	-.636
Freq;Own VS chrf	-.143	
Freq;Own VS DA	.571	N/A

Table 3: Pairwise Kendall's Tau correlations between our metrics and the standard metrics (DA for direct assessment). The columns are arranged the same way as in Table 2; the statistical significance pointed by asterisk is p < 0.05 (for positive tau values only). For 2022 data, we do not have DA scores, thus it is marked "N/A"; also the rankings by BLEU and chrf are same, thus the corresponding cells in 2022 are merged.

same most frequent initialization. The next level of correlation is for the pairs of different initializations with the same metric (F1 or our own, respectively); the lowest correlation is between the most distant variants (such as F1 with the first-occurrence initialization and our metric with the most-frequent initialization). Notably, such a clear trend can be seen only on the results of WMT2021 systems, while on 2022 data, the only clear correlation is between the F1 and our metric with first occurrence initialization. The detailed tau values are shown in Table 2. Looking back at Table 1, we can see that, for 2021 systems, the best ones are Facebook-AI, Online-Y, and Online-B according to any metric variant, and the worst are CUNI-Transformer and Online-G. As for 2022 systems, the best-rated ones are JDExploreAcademy, Lan-Bridge, CUNI-DocTransformer, and Online-B, while the worst-rated ones are CUNI-Transformer ALMAnaCH-Inria, Online-Y, and Online-G.

5.2 Comparison with Standard Automatic Metrics and Direct Assessment

We also wanted to compare our metrics to the traditional manual and automated evaluation approaches for MT. Unfortunately, the only published results of the considered MT systems were based on the evaluation of another dataset of news texts, see Akhbardeh et al. (2021) and the actual scores online.[3] However, they can still give us an approximate idea of the systems' relative performance. For the 2021 news track, we have both automatic scores (BLEU and chrf) and human direct assessment, while for the 2022 track, we only have the automated metrics, the same as for the previous year. To compare the rankings of our metric and the standard ones, we find it logical to use Kendall's tau correlation, as it was applied in previous metrics shared tasks Macháček and Bojar (2014). The results of this comparison can be seen in Table 3. Regarding the WMT2021 outputs, on the one hand, the correlation between any automatic metric and any of our variants is not as high (and the p-values do not show any significance). The correlation with direct assessment scores, on the other hand, is high (more than 0.6 on average), and shows also the statistical significance in 2 out of 4 cases (for F1 with both variants of pseudo-reference initialization).

Unfortunately, we cannot compare the 2022 results with human scores yet. For the 2022 automatic scores, the discrepancy between our metric and automated metrics is even bigger, which is represented by the negative τ value. If we analyze the ranking of the systems by the standard metric and of the proposed metrics, we can see that, for 2021, the tentative clustering into three groups (best-average-worst system) roughly coincides with the automated metrics, while for 2022 the general coincidence remains, but there are counterexamples such as Online-W which is best by BLEU

[3] https://github.com/wmt-conference/wmt22-news-systems

and chrf, and average by our metric. We can interpret the lack of correlation between the automated metrics and our metric the following way: the proposed metrics can give additional information compared to the dominant automated ones; moreover, they tend to correlate with the human document-level judgments, which are, as it has already been mentioned, more sensitive to the inconsistencies in translations on the document level.

5.3 Comparison of 2021 and 2022 Performance

The last notable comparison is the progress of systems that participated in both the 2021 and 2022 competitions; there were six such systems. We subtracted the 2021 scores from the 2022 scores and ranked the differences from the most significant increase to the biggest decrease. We did that both for our metrics and for the standard automatic ones. The first notable difference is that the changes in scores with our metrics are very small compared to BLEU and chrf, they are not bigger than 3% (while the smallest change in BLEU and chrf are 15% and 10%, correspondingly). Based on that, we may hypothesize that our metrics show that the system developers did not aim at increasing the term consistency of the translations. However, to check this hypothesis, we should analyze the architecture of the systems and possibly to compare their performance against the systems intentionally oriented at term preservation, such as Voita et al. (2019a). The detailed comparison of 2021 and 2022 algorithms is shown in Table 4.

6 Limitations and Perspectives

As was stated, we proceed with testing our metrics, both "extensively" (on more data) and "intensively" (by tweaking the inner parameters of the metric itself). Regarding the "extensive" analysis, we firstly should retrieve the automatic metrics obtained for the ELITR agreement corpus and compare them to our findings. Secondly, we should test our method on other language pairs or at least on the opposite English-to-Czech direction.

The second priority covers a more "intensive" analysis of the metric. The method that we suggest is based on several automated (or semi-automated) steps. For each of the steps (keyword extraction, word alignment, manual restriction of the term translations), different approaches and algorithms can be used. So far, we have tested the YAKE Campos et al. (2018) and KeyBERT[4] keyword extractors for the first step. We compared their performance on the legal text outside the main ELITR collection (this means that, for regex-based extractor, we created the templates based on the ELITR connection and applied it to the testing text). Tentative analysis shows that for the considered text, regex term extractor demonstrates the best performance, with 100% precision and 64% recall (7 out of 11 terms). Both YAKE and KeyBERT output an excessive number of false positive results, thus showing a dramatic decrease in precision (best performance – YAKE with 1-token keyword retrieval, 35%). The recall scores for these algorithms decrease as well: the comparable result is performed only by YAKE (54% for 1-token keyword retrieval), while the 2-token length YAKE shows 36% and KeyBERT shows 9%.

This can lead us to the conclusion that the regex term extraction is the best algorithm. However, if we apply these extractors to different texts of a similar domain – audit report (retrieved from another ELITR repository,[5]) we will see that the regex keyword extraction outputs no terms at all. The reason is that the terms in this report are introduced only in parentheses, with no additional explicit hints (such as "hereinafter referred as...") in the legal texts. Both machine learning-based algorithms, in contrast, manage to catch at least some of the necessary terms. This drives us to the conclusion, that for the robustness of the regex-based term extraction, we should take into account different "strategies" of introducing the terms in the document (sometimes – by parentheses, sometimes – by additional phrases). This means that, before evaluating a new collection of the exact text, we still need some human effort to understand the strategy of the term marking there. Another way for a bigger automatization can be using the combination of different keyword extraction algorithms, and choosing the terms through a majority vote or taking the union. Finally, we can look at the problem of the term extraction and alignment from an opposite perspective: if there is no reliable combination of the automated algorithms for these two steps, we can use our metric semi-manually: the steps 2-3 from Section 3 will be completely handed over to human annotators, and their results will be processed automatically

[4] https://maartengr.github.io/KeyBERT/index.html

[5] https://github.com/ELITR/wmt20-elitr-testsuite

	1st; F1	1st; Own	Freq; F1	Freq; Own	1st; F1 rank	1st; Own rank	Freq; F1 rank	Freq; Own rank	BLEU	chrf	BLEU rank	chrf rank
CUNI-Doc Transformer	.0006	.0012	.0006	.0013	3	3	3	3	.1603	.1109	5	4
Online-A	-.0065	-.0219	.0099	.0065	5	5	1	2	.1985	.1371	2	2
Online-B	.0223	.0395	-.0126	-.0097	1	1	5	5	.1749	.1197	3	3
Online-G	-.0005	.0006	-.0012	-.0049	4	4	4	4	.1533	.1031	6	6
Online-W	.0081	.0085	.0051	.0066	2	2	2	1	.2733	.1835	1	1
Online-Y	-.0399	-.0465	-.0288	-.0305	6	6	6	6	.1668	.1078	4	5

Table 4: Comparison of systems' progress from 2021 to 2022. The columns with the names of metrics (or the variants of our metric) denote the result of subtraction of the 2022 scores from 2021 scores. The "rank" columns sort the systems by their progress in the corresponding metric (1 - biggest increase, 6 - lowest increase/biggest decrease).

by steps 4-5. Of course, such implementation will be more time- and effort-consuming, but, firstly, it should still be faster than other manual evaluation approaches such as MQM, secondly, it will give us a model results of term extraction and alignment, against which we will compare the automated algorithms.

The last notable limitation of the proposed approach is rooted in linguistic issues. Although the legal texts are very consistent in using the same term for the same concept, there regularly appear cases of "legitimate" homonymy, where two terms can denote the same concept. This usually occurs when two or more antecedents can be referred to separately or by one term. The example is the following sentence: *X, hereinafter referred to as "Seller", and Y, hereinafter referred to as "Buyer", together also as "contracting parties"...*. Such ambiguity (when person X can be both referred as "Seller" and as "contracting parties") may cause the problems even within the correct translation, if in the original the chosen formulation would be "the Seller and the Buyer", and in the target language it would be chosen as "contracting parties". The current metric does not have any capacity to capture this feature of the legal language domain.

7 Conclusion

We have presented the metric for evaluating the terminology consistency of the automatically translated texts. Among its main advantages is its ability to be automatized and its relative simplicity of interpretation. We have tested our metric on the texts from the legal domain in the Czech-to-English translation pair, and we have obtained the results that, according to preliminary estimates, correlate with human document-level judgements and statistically differ from those of the automated metrics such as BLEU or chrF. We are continuing our analysis to understand the scope of our metric's functionality and test it on other language pairs.

We publish our code of the project online at the Github page[6] of the Institute of Formal and Applied Linguistics, Charles University. We will appreciate feedback on the current algorithm, and we are open to discussion and suggestions on its improvement.

8 Acknowledgements

This work was supported by GAČR EXPRO grant LUSyD (20-16819X, RIV: GX20-16819X) and we used services provided by the LINDAT/CLARIAH-CZ Research Infrastructure (https://lindat.cz), supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2018101).

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference

[6] https://github.com/ufal/wmt22-term-based-metric

on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Md Mahfuz Ibn Alam, Antonios Anastasopoulos, Laurent Besacier, James Cross, Matthias Gallé, Philipp Koehn, and Vassilina Nikoulina. 2021. On the evaluation of machine translation for terminology consistency. *CoRR*, abs/2106.11891.

Niyati Bafna, Martin Vastl, and Ondřej Bojar. 2021. Constrained decoding for technical term retention in english-hindi mt. In *Proc. 18th International Conference on Natural Language Processing, ICON 2021, December 16-19, 2021*. NLP Association of India (NLPAI).

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018. Yake! collection-independent automatic keyword extractor. In *Advances in Information Retrieval*, pages 806–810, Cham. Springer International Publishing.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Christian Hardmeier. 2012. Discourse in statistical machine translation: A survey and a case study. *Discours*, 11.

Matouš Macháček and Ondřej Bojar. 2014. Results of the WMT14 metrics shared task. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 293–301, Baltimore, Maryland, USA. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(4381):1–15.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Ferhan Ture, Douglas W. Oard, and Philip Resnik. 2012. Encouraging consistent translation choices. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 417–426, Montréal, Canada. Association for Computational Linguistics.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019a. Context-aware monolingual repair for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 877–886, Hong Kong, China. Association for Computational Linguistics.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019b. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1198–1212, Florence, Italy. Association for Computational Linguistics.

Tereza Vojtěchová, Michal Novák, Miloš Klouček, and Ondřej Bojar. 2019. SAO WMT19 test suite: Machine translation of audit reports. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 481–493, Florence, Italy. Association for Computational Linguistics.

Vilém Zouhar, Tereza Vojtěchová, and Ondřej Bojar. 2020. WMT20 document-level markable error exploration. In *Proceedings of the Fifth Conference on Machine Translation*, pages 371–380, Online. Association for Computational Linguistics.

Test Suite Evaluation: Morphological Challenges and Pronoun Translation

Marion Weller-Di Marco and Alexander Fraser
Center for Information and Language Processing
LMU Munich
{dimarco,fraser}@cis.uni-muenchen.de

Abstract

This paper summarizes the results of our test suite evaluation with a main focus on morphology for the language pairs English to/from German. We look at the translation of morphologically complex words (DE–EN), and evaluate whether English noun phrases are translated as compounds vs. phrases into German. Furthermore, we investigate the preservation of morphological features (gender in EN–DE pronoun translation and number in morpho-syntactically complex structures for DE–EN). Our results indicate that systems are able to interpret linguistic structures to obtain relevant information, but also that translation becomes more challenging with increasing complexity, as seen, for example, when translating words with negation or non-concatenative properties, and for the more complex cases of the pronoun translation task.

1 Introduction

Evaluating MT output is challenging. Document-levels metrics give a rather coarse-grained estimation of the overall translation quality, but cannot determine how well a system operates for particular challenges. Translations do not have a deterministic solution, but there are always several possibilities for a valid translation, making a focused evaluation of particular phenomena difficult.

The annual WMT Shared Task provides the possibility to submit custom test suites to be translated in addition to the regular test sets, which allows the investigation of the translation performance of state-of-the-art systems when presented with particular translation tasks. In this test suite, we focus on morphological challenges for English to/from German translation: For German–English translation, we look at the translation of morphologically complex words, in addition to a small set of sentences where a subtle difference (singular vs. plural) needs to be detected. For English–German, we study how complex noun phrases are translated – as compounds or rather as multi-word phrases. Furthermore, we add a pronoun translation task and evaluate the translation of the English pronoun *it* into its German equivalents *er/sie/es*, depending on the gender of the noun it refers to.

The test suite does not aim at measuring a system's general translation performance – this is already assessed by means of a manual evaluation and various other metrics in the main shared task – but rather at evaluating the translational behaviour for carefully selected words or phrases. As the sentences in the test suite are not parallel, we opt for a semi-automatic approach where translation options for the words in question are manually collected and then matched with the translation output. Thus, only the translation of the relevant word is considered, whereas the rest of the sentence is ignored.

2 Data Creation and Evaluation

In the following, we outline the process of composing and evaluating the test suite.

Selection of words The sets for the analysis of translating morphologically complex words, compound variants and compounds for re-translating into English are mostly based on a word-frequency list from DeWac[1] (Baroni et al., 2009). The words were morphologically analyzed with SMOR (Schmid et al., 2004). Based on this analysis, words for the aforementioned categories were selected:

- Morphologically complex words: Words with a high degree of complexity and properties such as different forms (e.g. with/without *Umlaut*) in stem and derivations; with negation prefixes or particles or verbal components.
- Compound variants: compounds for which both variants NN1 NN2 and NN2 NN1 exist.
- Compounds for re-translation: adjectives and nouns with up to four components.

[1] https://wacky.sslmit.unibo.it/doku.php?id=frequency_lists

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 458 - 468
December 7-8, 2022 ©2022 Association for Computational Linguistics

Sentence selection We manually retrieved sentences containing the selected words, using Google and the search function provided by the corpus platform DWDS (Geyken et al., 2017) with the corpus *Webmonitor*[2] which is daily updated. The search was (mostly) restricted to newspaper entries from this year, in order to obtain "new" data that was not previously seen in the (monolingual) training data.

Evaluation We identified the translation hypotheses of the relevant words using word alignment (Eflomal (Östling and Tiedemann, 2016)), which were then matched with a manually composed lexicon containing translations options. This step is semi-automatic in the sense that yet unseen translation options need to be verified and added to the lexicon. For the verification, we took into account the sentence context. Being mainly interested in adequacy (i.e. reproducing the meaning of the source word) we allowed for some leeway at the level of fluency, which is difficult to determine anyway in sentences that are not always fully grammatical.

3 DE–EN Translation

This section summarized the design and the outcome of the four categories in DE–EN translation.

3.1 Morphologically Complex Words

We are interested in the translation of morphologically complex words that contain interesting morphological properties such as negation, particles, verbal elements or non-concatenative derivation, which often pose a challenge for translation.

Consider, for example, the word *abrissunwillig: abreißen + un + willig* (*tear down + un + willing*: *unwilling to tear down*), which consists of a nominalization (*abreißen*$_\text{V}$ $\rightarrow$ *Abriss*$_\text{N}$), a negation prefix (*un-*) and an adjective (*willig*: *willing*). In addition to being complex, there is also a non-concatenative operation in the derivation, namely the stem change in *abriss-* vs. *abreißen*. This makes it difficult for linguistically uninformed splitting approaches to find a segmentation into meaningful splits that match with, and thus benefit from, other instances of related words with the same stem.

Many of the selected words emerged from creative use of language and are rather low-frequency. This is to challenge the systems to analyze the words rather than having them already memorized. The words can be loosely grouped as follows:

		JDExplore Academy	Lan-Bridge	LT22	Online-A	Online-B	Online-G	Online-W	Online-Y	PROMT
NEGATION	correct	49	51	15	50	55	47	54	51	49
	incorrect	8	6	42	7	2	10	3	6	8
	→ polarity	–	4	9	1	2	4	–	3	1
	→ lex.	8	2	30	6	–	6	3	3	7
	→ untransl.	–	–	3	–	–	–	–	–	–
VERB	correct	12	13	2	12	15	13	14	11	11
	incorrect	4	3	14	4	1	3	2	5	5
UML	correct	50	49	18	48	49	43	50	43	40
	incorrect	2	3	34	4	3	9	2	9	12
NON CONC	correct	43	40	5	28	44	35	36	34	24
	incorrect	8	11	46	23	7	16	15	17	27
PART INF	correct	62	64	24	63	62	64	64	60	58
	incorrect	14	12	52	13	14	12	12	16	18

Table 1: Morphologically complex words.

Negation: words containing the negation morphemes *un-* (*unschmelzbar*: *unmeltable*) or *-los* (*knopflos*: *without buttons*). We are in particular interested how the negation is realized, i.e. as an isomorphic, word-internal negation vs. word-external negation. This group comprises 57 sentences.

Verbal elements: the form of verbal elements in derivations often differs from that of the verb stem (*aufbruchsbereit*: *ready to go*; *aufbrechen*: *to leave*). This group comprises 16 sentences.

Stem change (Umlaut): words containing an *Umlaut* in the derivation but not in the stem: *blümchenbedruckt/Blume* (*printed with little flowers/flower*). This group comprises 52 sentences.

Non-concatenative words: adjectives derived from nouns with non-concatenative properties, e.g. *langwimprig/Wimper*: *long-lashed/lash*. This group comprises 51 sentences.

Complex words: words containing particles, such as *mitzittern* (lit: *tremble-with*; *to sympathize, share somebody's emotions*) and words containing *-zu-* infixes (e.g. *aufzutürmen*: *to stack up*). Words of this group are often difficult to translate directly. This group comprises 76 sentences.

Table 1 gives an overview for all five categories. For words containing **negation**, we find that most systems made between 6 and 10 errors (out of 57), with three systems being much better or worse.

For the errors, we distinguish between *lexically incorrectly* translated and *wrong polarity*[3]. For the lexically bad translations, we found that a majority still contained a negation morpheme (such as

[2] https://www.dwds.de/d/korpora/webmonitor

[3] The negation is incorrectly reproduced in the translation, either through omission or by a word of the opposite meaning.

nuancen**los**	nuanced (3), nuances (1)
manövrier**un**fähige	maneuverable (3),manoeuvrable (1), manoeuvring (1)
familien**un**freundliche	family-friendly (1)
datenschutz**un**freundliche	data protection-friendly (2), data-protection-friendly (1)
keim**un**fähig	viable (1), germinate (2)
kunden**un**freundlichen	client-friendly (1)
klima**un**freundliches	climate-friendly (1)
fahrrad**un**freundlichste	most bicycle-friendly (1)
unverblasst	still faded (1)
unaufgetaut	unfrozen (1)
unadeligen	aristocratic (1)
kalorien**lose**	calories (1)

Table 2: Translations with wrong polarity (all systems).

knopflos (*buttonless*) → *headless*). For translations with wrong polarity, we observed that in particular words with an infix negation morpheme are error-prone, especially when considering that the test set contains only 13 sentences with such words. Table 2 lists the (otherwise lexically correct) translations with wrong polarity.

Among the correct translations, we observe the entire range between no translation variation (e.g. *unwählbar* ↔ *unelectable* and *vorwarnungslos* → *without warning*) and lexical and local structural variation, as shown in table 3.

For words containing **verbal elements** that differ from the lemma of the verb, the systems' performances range from nearly all correct to nearly all incorrect. For this subset, there was no clear trend of error, certainly also due to its small size. One thing that we observed was that for *abrissbedroht* (*threatened by demolition*), *abrissbereit, abrissreife* (*ready to be demolished*), *abrissgeweihten* (*marked for demolition*), *abrisswilligen* (*willing to demolish*) a common mistranslation was just *demolished*, even though the state of being actually demolished is not described by any of these words.

The words with a **stem change (Umlaut)** lead to mixed results; for the words with **non-concatenative properties**, we observe even more errors. Among the incorrectly translated words, there is a tendency that the part with the non-concatenative properties is mistranslated, whereas the other, more easy part, is correct (cf. table 4). Finally, the words containing **particles or infixes** were challenging to translate, even though some words were considerably more difficult. In particular, the set included some verbs that cannot be translated isomorphically. One example is the combination of *kaputt* (*broken*) + *verb*, in analogy to *kaputtmachen* (*to break*, lit. *kaputt-make*): *kaputtsparen* (*to destroy through excessive money saving*) or *kaputtsanieren* (*to destroy through excessive renovating*). With the exception of *kaputtschlägt* and *kaputtzukriegen* (*to break*), they were nearly always translated incorrectly. In particular *kaputtsparen* was often translated as *saved from damage* or similar, the opposite of the intended meaning. In contrast, for *schönreden* (*to gloss over, to sugar coat*, lit: *beautiful* + *talk*), a generally similar construction, about half of the translations were correct.

quittungslos	without receipt (9), without receipts (2), without a receipt (2), receiptless (1), receipt-free (1)
nuancenlos	without nuances (4), nuanceless (3), nuance-free (3),nuance-less (2), unnuanced (1), lacking in nuance (1)
unaufgetaut	unthawed (3), without thawing (1), without defrosting (1), undefrosted (1), before it is thawed (1)
unverblasst	unfaded (2), still vivid (1), not yet faded (1), not faded (1), still fresh (1)

Table 3: Translation variants for words with negation morphemes (only correct translations shown).

	correct	incorrect
langwimprigen	long-lashed (4)	long-drawn (1), long tail (1), long-winded (1), long-eyed (1) long-wimprigen (1)
sonnenbebrillt	in sun glasses (2), with sunglasses (1), wearing sunglasses (1)	bespectacled by the sun (1), in the sun (1), sunglassed (1)
löwenmähnige	lion-maned (15)	lion-eyed (1) lion-like (1), duel-like (1)

Table 4: Translating non-concatenative words.

3.2 Compound Variations

Compounds are commonly occurring in German and their translational behaviour has been studied extensively. An important aspect in compound translation is to correctly reproduce the relation between the head and modifier in noun-noun compounds, which we aim to investigate in this category by looking at compound pairs that consist of the variants NN1 NN2 and NN2 NN1, such as *Oliven|öl* and *Öl|olive* (*olive oil* vs. *oil olive*) or *Leder|stiefel* and *Stiefel|leder* (*leather boot* vs. *boot leather*).

The compound variants NN1 NN2 and NN2 NN1 have different heads and thus a different meaning (as opposed to variation in hyponymy/hypernymy) and are not generally interchangeable[4]. We thus

[4] We found, however, that in some cases, there is an acceptable one-word translation for both variants, e.g. *Absatzschuh*

	JDExplore Academy	Lan-Bridge	LT22	Online-A	Online-B	Online-G	Online-W	Online-Y	PROMT
correct	57	58	29	59	59	60	59	59	57
wrong order	–	–	2	–	1	–	–	–	–
head missing	1	–	7	–	–	–	–	–	1
mod missing	–	–	5	1	–	–	1	–	–
bad transl.	2	2	17	–	–	–	–	1	2

Table 5: Translation results of compound pairs. *wrong order*: wrong order of head and modifier; *head/mod missing*: only translated head or modifier; *bad transl*: translation was either missing or wrong.

retrieved different sentences for each variant: this means that the compound variants are not analyzed in a minimal pair setting, but each variant is presented in an appropriate and natural context.

This category is somewhat inspired by one of the error types introduced by Sennrich (2017), where translation probabilities for contrastive sentences containing compound variants (a correct vs. a wrong translation consisting of a compound with switched components) are compared.

Table 5 shows the results for 15 compound pairs, with 2 examples per variant in most cases, resulting in 60 sentences total. All systems, with the exception of one, translated most compounds correctly. Furthermore, there is no dominant error type for cases with incorrect translation. This indicates that through most systems, there is a generally good understanding of compound structure and subsequent translation, even in cases such as the high-frequency *Olivenöl* (21M google hits[5]) vs. the low-frequency *Ölolive* (507 google hits).

3.3 Compound Translation

In this section, we look at the translation of compounds consisting of two to four components. This set of compounds[6] also serves as a basis for the experiment in section 4.1 which studies how English noun phrases are translated into German.

This word set contains some "newish" words, i.e. words that are not new per-se, but became considerably more frequent recently, such as *Gasengpass* (*gas bottleneck*) and some Covid-related terms such as *Impfbereitschaft* (*willingness to be vaccinated*). Most words in this set are compositional, and very few are non-compositional compounds, such as *Dornröschendasein* (*Sleeping Beauty existence*).

	JDExplore Academy	Lan-Bridge	LT22	Online-A	Online-B	Online-G	Online-W	Online-Y	PROMT
correct	68	70	34	70	70	71	69	68	71
wrong	5	3	35	3	3	2	3	5	2
missing/copy	–	–	4	–	–	–	1	–	–

Table 6: Results for translating compounds into English.

Table 6 shows the results for translating compounds into English; most systems did quite well. Among the compounds with the most consistent translations are *Sonnenblumenkernöl* (*sunflower seed oil*) and *Haifischflossensuppe* (*shark fin soup*), i.e. compounds with a straightforward literal translation. Similarly, the somewhat new *Testmüdigkeit: test fatigue (17), testing fatigue(1)* (occurrences in two sentences) leads to consistent translations. One of the more difficult words was *distanzlernende (distance learning)*, which 5 of the 9 systems translated correctly. The incorrect translations did not quite capture the meaning and translated into *learning distance*, *learning about distance* and *to dance* (probably due to an incorrect splitting that contained the German "tanz").

3.4 Preserving Morphological Information in Syncretic Forms

Understanding the precise meaning of a word and its function in the sentence is crucial to obtain a good translation. This includes the comprehension of relevant morphological features.

While German is rich in different inflected forms, there is also a certain degree of syncretism (forms with different morphological features sharing the same surface form). For example, *Hund* (*dog*) can be dative, accusative and nominative, *Unternehmen* (*company*) can be singular and plural. Usually, this can be resolved by the context, often by means of the determiner: *dem*$_{\text{DAT}}$/*den*$_{\text{ACC}}$/*der*$_{\text{NOM}}$ *Hund* and *das*$_{\text{SG}}$/*die*$_{\text{PL}}$ *Unternehmen.*

In this experiment, we look at number in non-subject words as (i) number is the only feature of nominal inflection that is shared between German and English, and (ii) there are no further ramifications to the rest of the sentence. We designed a setting in which the disambiguating context, a definite article, is not directly adjacent to the word in question, but separated by an inserted phrase.

→ *heel, heeled shoe* and *Schuhabsatz* → *heel, shoe heel*.

[5] Search of the citation form in double quotes. Numbers reported by Google give only a rough idea of the true frequency on the web, but are sufficient to estimate the order of magnitude.

[6] This set of words is not exactly the same as in section 4.1.

Die Verzögerungen sind auf Engpässe bei **den** mit der Umsetzung beauftragten **Unternehmen** und auf ... zurückzuführen. The delays are to bottlenecks at **the** with the implementation charged **companies** and to ... due
Die Verzögerungen sind auf Engpässe bei **dem** mit der Umsetzung beauftragten **Unternehmen** und auf ... zurückzuführen. The delays are to bottlenecks at **the** with the implementation charged **company** and to ... due
The delays are due to bottlenecks at **the companies/company** charged with the implementation and to ... .

Table 7: Example for minimal sentence pairs.

	JDExplore Academy	Lan-Bridge	LT22	Online-A	Online-B	Online-G	Online-W	Online-Y	PROMT
Correct	15	16	8	15	16	17	17	18	17
Incorrect	3	2	7	3	2	1	1	–	1
NA	–	–	3	–	–	–	–	–	–

Table 8: Preserving number information: *Correct*: the noun in singular and plural was translated correctly. *Incorrect*: for at least one noun, the number was incorrect. *NA*: not translated or otherwise impossible to judge.

We created 18 minimal sentence pairs with the only difference being a singular vs. a plural article, in order to test whether the noun (with identical forms in both sentences) is correctly translated. The sentences contain "nested prepositional phrases" where an inserted prepositional phrase separates the article and the noun, cf. table 7.

Table 8 shows the results for the task of preserving number information: most systems can handle this problem reasonably well, indicating that the systems have the ability to interpret the sentence structure and to identify the relevant context.

4 EN–DE Translation

In this section, we look at re-translating compounds and present a pronoun translation task.

4.1 Compound Creation

To prepare the test set, we translated the German compounds from section 3.3 into English, including structural or lexical variations if possible (cf. table 9 for some examples) and retrieved English sentences with these translations, resulting in a set of 102 sentences. We distinguish between "phrase" (PHR), containing a preposition (such as *interpreter for sign language*) and "compound" (COMP) where the order of the words corresponds to a compound (as in *sign language interpreter*).

The results in table 10 show a tendency to keep the structure, i.e. translating a compound-like structure into a compound, and a phrase into a phrase rather than a compound, even though there are differences depending on the word.

Gebärdensprach-dolmetscher	sign language interpreter interpreter for sign language
Obstbaumschnittkurs	fruit tree pruning workshop workshop on fruit tree pruning
Kleinkläranlagen-betreiber	small wastewater treatment plant operators; operators of small wastewater treatment plants
Kreuzworträtselfrage	crossword question crossword puzzle question

Table 9: Structural and lexical variants in the compound translation task.

	JDExplore Academy	Lan-Bridge	Online-A	Online-B	Online-G	Online-W	Online-Y	OpenNMT	PROMT
Comp → Comp	58	56	60	57	58	55	47	49	55
Comp → Phr	13	15	10	11	11	17	11	13	6
Phr → Comp	5	9	2	9	5	7	2	3	4
Phr → Phr	23	18	24	18	23	21	25	18	19
Wrong transl.	2	3	6	6	5	2	16	19	18
Copied EN	1	1	–	1	–	–	1	–	–

Table 10: Translating English complex phrases.

For example, the variants *wearers of headscarves* and *headscarf wearers* were mostly translated by the compound *Kopftuchträger(innen)*, with only two instances of *Träger von Kopftüchern*. In contrast, both *pacemaker wearer* and *pacemaker carrier* have a more equal distribution of *Träger von Herzschrittmachern* and *(Herz)Schrittmacherträger*. A more complex example, *willingness to get vaccinated*, was translated to the corresponding compound *Impfbereitschaft* (6 times), as *Bereitschaft zur Impfung* (3 times) and *Bereitschaft, sich impfen zu lassen* (9 times). The variant *unwillingness to get vaccinated* proved more problematic: only three systems obtained correct translations: *Impfunwilligkeit* (2) and *Unwilligkeit, sich impfen zu lassen* (1). The translation *Impfunbereitschaft*, while transporting the correct message, is questionable. In the remaining 5 cases, the negation was ignored.

4.2 ContraCat: Translating Pronouns

The translation of pronouns is often more difficult than it seems at a first glance: a translation system requires diverse linguistic information to produce a

1	The mouse ate the cookie and the **bear** drank the milk. **It** drank the milk quickly.
2	The **tiger** ate the ice cream. **It** was happy.
3	The giraffe ate the **steak**. **It** was cooked.

Table 11: Examples for the ContraCat template set.

target-language pronoun with the correct morphological features such as gender, number or case.

To translate the English *it* into German, a translation system needs to identify the noun *it* refers to and to have knowledge about that noun's gender[7] in German, as illustrated below:

... a dog ... it ... → *... ein Hund*$_{\text{MASC}}$ *... er ...*
... a cat ... it ... → *... eine Katze*$_{\text{FEM}}$ *... sie ...*
... a zebra ... it ... → *... ein Zebra*$_{\text{NEUT}}$ *... es ...*

To analyze the translation of pronouns, we make use of the template set *ContraCat* (Stojanovski et al., 2020) which consists of sentence pairs where several nouns are introduced in the first sentence, and a pronoun *it* in the second sentence either refers to one of these nouns, or is generic as in *it is raining*. The sentences are constructed in a way that the relevant noun/context can be derived through either world knowledge or through analyzing the structure of the sentence. Furthermore, the sentences are designed such that the nouns of e.g. the two subjects (mouse and bear in sentence 1 in table 11) have translations into German with different genders (*Maus*$_{\text{FEM}}$ and *Bär*$_{\text{MASC}}$) in order to allow for an unambiguous evaluation.[8] Table 11 shows three examples; an overview of all template categories can be found in table A.

Technically, each sentence consists of two short sentences. As this might be disadvantageous in some system settings, we generated a second version where we joined the two short sentences with "and" into one sentence. In the evaluation, these variants will be referred to as 2S and AND.

In its original form, the template set provides three translation hypotheses, each with a different translation option (male/female/neutrum) for *it*, for which the system's likelihood to produce the correct translation is then measured.

To be used in an actual translation scenario, we adapt the evaluation process: given the template structure, we first identify the antecedent (the noun that is referenced by the pronoun *it*), and then its translation and the translation of the pronoun *it* in the target sentence using word alignment (Eflomal, Östling and Tiedemann (2016)). The translation options of the nouns observed in the different systems' outputs are listed in a manually compiled dictionary[9], alongside their German grammatical gender. With this, the translated pronoun can be automatically matched with the noun's gender.

4.2.1 Test Set Creation

From the original test suite[10], we randomly selected 100 sentences for each of the 20 categories (cf. table A for an overview), with the exception of the category *world knowledge*, for which 200 sentences were selected as this category comprises the scenario of addressing an animate noun (animal) vs. inanimate noun (food). Doubling the sentences for the AND variant results in 4200 sentences total.

4.2.2 Evaluation and Results

Table 12 shows the results of translating pronouns. For the categories *event_** and *pleo_**, where the translation *it* → *es* is always expected, nearly all systems have a perfect score. The other categories where the antecedent needed to be derived from the context are more challenging, however without a clear pattern between the systems. We can observe two tendencies, even though not consistent through all systems: first, the variant AND often leads to better results, probably due to the fact that sentences are often the "standard unit" for translation, whereas the two sentences in variant 2S might be considered separately, depending on the systems' architecture. Second, sentences where the antecedent is the second NP of the first sentence, i.e. closer to the *it*, tend to get better results.

Looking further into the errors, we find that *es*$_{\text{NEUT}}$ is often preferred over a a feminine or masculine form. This might simply be the case because *it* → *es* is the default translation, and also because the generic *es* can oftentimes be considered grammatical, even though a translation into the gender-specific pronoun would be possible.

A general problem with this template approach is the degree of freedom in the translation process: sometimes the pronoun is just not translated (cf. table 13), and in some cases, it is possible to formulate the sentence such that the pronoun *es* leads to

[7] Further features leading to variations at the level of grammatical case and number will be ignored here.

[8] This was guaranteed for the pre-defined translations in the original setting. In actual translations, there can be more variation, for example *deer* → *Hirsch*$_{\text{MASC}}$, *Reh*$_{\text{FEM}}$, *Wild*$_{\text{NEUT}}$.

[9] The dictionary comprises entries for 141 English nouns, with one to four translation options.

[10] https://github.com/BennoKrojer/ContraCAT

	JDExplore Academy		Lan-Bridge		Online-A		Online-B		Online-G		Online-W		Online-Y		OpenNMT		PROMPT	
	2S	AND	2S	AND	2S	AND	2S	AND	2S	AND	2S	AND	2S	AND	2S	AND	2S	AND
event_chaos	100	100	100	100	100	100	100	100	100	100	100	99	100	100	100	100	100	100
event_happened	100	100	100	100	100	100	100	100	100	100	100	99	100	100	99	100	100	100
event_situation	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
event_surprise	100	100	100	100	100	100	100	93	100	100	100	99	100	100	100	100	100	100
gender_step	95	92	22	95	21	84	22	92	21	91	88	97	21	92	80	66	22	89
obj_drink	77	96	22	95	18	57	22	76	35	50	81	81	18	30	26	38	29	27
obj_eat	1	27	37	6	23	26	37	26	32	38	42	10	23	24	14	23	20	28
obj_verb_drink	100	100	20	98	16	53	20	95	38	53	84	92	22	57	24	41	18	41
obj_verb_eat	1	1	36	2	25	30	33	11	29	36	43	8	26	32	2	25	28	41
pleo_believe	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
pleo_rain	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
pleo_seem	100	100	100	100	100	100	100	100	100	100	99	99	100	100	100	100	100	100
pleo_shame	5	100	100	100	100	100	100	100	100	100	100	100	100	100	97	100	100	100
subj_drink	36	45	34	34	34	34	34	37	34	34	45	84	34	34	85	68	34	34
subj_eat	45	44	45	45	45	45	45	45	45	45	44	24	45	45	18	35	45	45
subj_verb_drink	99	99	34	100	34	94	34	100	34	98	100	100	34	75	100	93	34	64
subj_verb_eat	22	20	34	20	34	11	34	13	34	34	54	2	34	22	0	14	34	25
verb_drink	100	100	24	98	22	72	25	90	26	68	84	96	22	72	71	19	22	43
verb_eat	2	2	27	2	18	18	26	11	25	33	43	20	18	21	28	8	21	45
world_knowl.	180	194	77	200	71	124	73	195	67	133	185	181	76	126	97	77	77	97

Table 12: Results for pronoun translation using the ContraCat template, the number indicates the amount of correct pronoun translations (out of 100 for all except for world_knowledge, which has 200 test sentences). (Note: in JDExploreAcademy–pleo_shame, *it is a shame* is nearly always translated as *"Schade."*, i.e. without a pronoun.)

The mouse ate the cookie and the **sheep**$_{\text{SG/PL}}$ drank$_{\text{SG/PL}}$ the tea. **It**$_{\text{SG}}$ liked the tea.
Die Maus aß den Keks und die **Schafe**$_{\text{PL/NT}}$ tranken$_{\text{PL}}$ den Tee. Er$_{\text{SG/MASC}}$ mochte den Tee.

Table 13: Example for incorrectly passed-on number.

The cow ate and the dog drank. It drank a lot.
Die Kuh aß und der Hund trank viel. *The cow ate and the dog drank a lot.*
The frog ate the fruit and **it** had a sour taste.
Der Frosch aß die Nuss und ∅ hatte einen sauren Geschmack. *The frog ate the fruit and had a sour taste.*

Table 14: Examples for pronoun omission.

a grammatical sentence. For example, *the -animal- liked it* (*it* →*food item*) can be translated as *Dem -Tier- gefiel/schmeckte es*. This is a valid translation, even though not strictly in the sense of the intended meaning as in *dem -Tier- schmeckte er (*→ *Apfel*$_{\text{MASC}}$*)* vs. *dem -Tier- schmeckte sie (*→ *Banane*$_{\text{FEM}}$*)* . For the sake of evaluation, we count a translation only as correct if the pronoun exists and matches in gender with the noun it refers to.

While this experiment only focused on *gender*, we also observed some cases that extended to *number*, namely in a few cases where the English singular and plural forms are the same. In the example in table 13, the number of *sheep* is not directly visible in the first part of the sentence, but can be disambiguated through the singular form *it*. The translation contains *Schafe* in plural, but *er* as translation of *it* is singular/masculine (*Schaf* is neutrum).

5 Related Work

The linguistic and morphological compentence of translation systems is a topic of previous and ongoing research. Isabelle et al. (2017) present a challenge set for English to French translation targeting linguistic divergence between the two language pairs. Their hand-crafted set has a focus on morpho-syntactic, lexico-syntactic and syntactic divergences. Burlot and Yvon (2017) present an analysis of minimal pairs representing a contrast that is expressed syntactically in EN and morphologically in a morphologically rich language (DE, CZ and LV). For a source test sentence (the base), variant(s) containing exactly one difference with the base (e.g. person/number/tense of a verb or number/case of a noun/adjective or polarity) are generated and automatically evaluated, counting a translation as correct if the targeted feature is produced correctly in the target language. The work of Burchardt et al. (2017) and Avramidis et al. (2019) comprises the DFKI test suite for German to English MT. Their test set consists of over 5k sentences to analyze over 100 categories, including negation, composition, function words, subordination, non-verbal agreement, multi-word expressions, verb tense/aspect/mood, lexical ambiguity and punctuation. LingEval97 (Sennrich, 2017) is a large-scale data set of 97000 contrastive English–German translation pairs where errors (on the level of agreement, auxiliaries, verb particles, polarity and swapped compound components) haven been automatically created. It is then measured whether a reference translation is more probable than the corresponding contrastive translation containing an inserted error. An obvious question with this approach is whether forced translation mimics the MT system's "natural behaviour", i.e. whether the presented sentence e is the system's best choice given the source sentence f. This question is addressed in Vamvas and Sennrich (2021) where it is argued that test data should be chosen such that there is minimal discrepancy between the training data and the data to be evaluated. They recommend that test sentences be created from machine generated text rather than using human-written references. The paper proposes an updated version of LingEval97.

6 Conclusion and Future Work

This paper summarizes the results of our WMT22 Test Suite, looking at the translation of morphologically complex words, compounds and a set of minimal pairs to assess the preservation of number. Our evaluation shows that on one side, the translation of morphologically complex words is not without challenges, in particular for low-frequency words and when containing negation. On the other hand, the handling of the (structurally much simpler) compounds NN1 NN2 vs NN2 NN1 and the preservation of the number feature worked quite well. The results for the pronoun translation experiment were mixed.

Our results indicate that the systems have a generally good understanding of linguistic structures, but also that at a certain degree of (morphological) complexity, problems start to arise. For research in MT, this means that modeling morphology, particularly negation and non-concatenative processes, might be worthwhile.

The test suite, with the exception of the pronoun translation task, is based on a manually created set of sentences alongside matching dictionaries. While this has the advantage of presenting the selected words/phrases in a natural context, it comes with a comparatively high amount of manual effort, making it difficult to upscale. In contrast, the artificial data used in the pronoun translation task allows for a comparatively straightforward evaluation, but sounds unnatural and likely differs considerably from the MT training data, which might even bias the results to a certain extent.

For future work, we intend to look into the generation of meaningful sentences with particular properties that allow for a systematic evaluation of MT.

Limitations

There are several limitations to this work: first, the work is obviously limited in terms of data-set size and the small number of language pairs considered. As there is a certain amount of manual selection and annotation required, this is generally a tricky problem to address. As mentioned previously, we plan to work on more sophisticated test data generation as a basis for a more focused evaluation. Another limitation is a lack of generalizability: the presented analyses offer only partial insights and provide but a first glimpse into understanding to what extent morphological information is captured and passed on in machine translation.

Ethics Statement

We have no ethical concerns about the research presented in this paper. The data selection and annotation work was carried out by the first author.

Acknowledgements

This work was supported by the DFG (grant FR 2829/4-1).

References

Eleftherios Avramidis, Vivien Macketanz, Ursula Strohriegel, and Hans Uszkoreit. 2019. Linguistic evaluation of German-English machine translation using a test suite. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers)*, pages 445–454, Florence, Italy. Association for Computational Linguistics.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Aljoscha Burchardt, Vivien Macketanz, Jon Dehdari, Georg Heigold, Jan-Thorsten Peter, and Philip Williams. 2017. A linguistic evaluation of rule-based, phrase-based, and neural mt engines. *The Prague Bulletin of Mathematical Linguistics*, 108.

Franck Burlot and François Yvon. 2017. Evaluating the Morphological Competence of Machine Translation Systems. In *Proceedings of the Second Conference on Machine Translation*, pages 43–55, Copenhagen, Denmark. Association for Computational Linguistics.

Alexander Geyken, Adrien Barbaresi, Jörg Didakowski, Bryan Jurish, Frank Wiegand, and Lothar Lemnitzer. 2017. Die Korpusplattform des „Digitalen Wörterbuchs der deutschen Sprache“ (DWDS). *Zeitschrift für Germanistische Linguistik*, 45(2):327–344.

Pierre Isabelle, Colin Cherry, and George Foster. 2017. A challenge set approach to evaluating machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2496, Copenhagen, Denmark. Association for Computational Linguistics.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *Prague Bulletin of Mathematical Linguistics*, 106:125–146.

Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. SMOR: A German Computational Morphology Covering Derivation, Composition, and Inflection. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, pages 1263–1266, Lisbon, Portugal.

Rico Sennrich. 2017. How Grammatical is Character-level Neural Machine Translation? Assessing MT Quality with Contrastive Translation Pairs. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 376–382, Valencia, Spain. Association for Computational Linguistics.

Dario Stojanovski, Benno Krojer, Denis Peskov, and Alexander Fraser. 2020. ContraCAT: Contrastive coreference analytical templates for machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4732–4749, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jannis Vamvas and Rico Sennrich. 2021. On the limits of minimal pairs in contrastive evaluation. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 58–68, Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Appendix

Words with negation: *abgaslose, abgaslosen, akzentlosen, datenschutzunfreundliche fahrradunfreundlichste, fahrunfähig, familienunfreundlich, familienunfreundliche, flugunfähigen, handlungsunfähig, kalorienlose, keimunfähig, klimaunfreundliches, knopflosen, kundenunfreundlichen, lückenlose, manövrierunfähige, nuancenlos, quittungslos, Rücksichtlose, tageslichtlose, unabgefüllten, unabgeschirmt, unablösliche, unadeligen, unanfechtbar, unanfechtbare, unangemeldete, unaufgetaut, unausgereift, unauswechselbar, unbegehbar, unfußballerisch, ungepflügten, ungeschliffen ,ungeschliffene, unkontrollierbare, unliebenswert, unregierbaren, unreparierbar, unreparierbarer, unsanierten, unschmelzbaren, unverblasst, unverderblich, unverhangene, unverwundbaren, unwählbar, unzerknittert, unzerschnittene, vorwarnungslos*

Words with verbal element: *abrissbedroht, abrissbereiten, abrissgeweihten, abrissreife, abrisswilligen, abwieglerisch, aufbruchsbereiten, aufbruchsicher, aufwieglerisch, aufwieglerischen, ausbruchsartigen, ausbruchsicher, ausstiegswillige, bestbesprochenen, weitergesponnen*

Words with Umlaut: *ananasförmigen, Anemonenblütige, barhändig, blümchenbedruckte, blümchenbedruckten, blümchentapetigen, doppelbödig, doppelköpfig, doppelköpfigen, einblättrig, einblättrigen, einsträngig, einsträngige, einsträngigen, engräumig, fädenziehende, fältchenmindernden, gehirnwäscherische, gehirnwäscherischen, großäugig, großäugigen, großräumig, höhergeschossigen, höherrangiger, höherwüchsiger, hundertäugigen, hütchenförmige, kaltblütig, kannenförmige, kannenförmiges, kleinräumig, kurzfädige, kurzfädigen, pünktchenförmig, rot-schnäblige, Rundbäuchig, rundbäuchige, sanftäugige, sanftäugigen, schnellfüßige, schnellfüßigen, spitztürmige, städtebauliche, städtebaulichen, städteübergreifend, täschchenlosen, viersträngig*

Words with non-concatenative properties: *angsthasig, aprilwettrig, dreistreifig, dunkelschalige, dünnschalig, dünnschalige, eigenpfotig, einhöckrigen, einstreifig, engmaschig, erdbeerartigen, flinkfingrige, flinkfingriger, grobbrockige, grobmaschig, grobmaschigen, grobmaschiger, großfenstrigen, großmaschig, großnasigen, hellschalig, hochgiebligen, hornbrilligen, langwimprigen, leichtpfotig, löwenmähnige, rotschalig, rotwangige, samtpfotigen, schmalhüftige, schnarchnasig, sonnenbebrillt, spitzgiebligen, Unbebrillt, zartschalig, zweihöckrige, zweistreifig*

Words with particle/*zu*-infix : *anföhnen, angeföhnt, aufdimensioniert, aufeinandergestapelt, aufgetürmt, aufgetürmten, auftürmen, aufzutürmen, beschuhten, coronabedingter, dichtgedrängten, eingerahmte, eingeschnürt, einrahmende, einschnürende, erdzugewandten, Fehlbefüllte, Fehlbefüllung, fehlbesetzt, fehlgeleitete, Fehlübersetzung, feindosiert, feindosierte, feingekleidete, fernsteuerbar, fernsteuerbarer, fertiggepackten, festbetoniert, festgerostet, festgeschraubt, geheimgehaltene, geheimzuhalten, geheimzuhaltenden, gutriechende, heißbegehrter,hitzebedingt, hochaufgetürmte,* *hochbeschuhten, kaputtanalysiert, kaputtgespart, kaputtgestanden, kaputtsanieren, kaputtschlägt, kaputtzukriegen, kaputtzusparen, kontinentübergreifende, kostümbedingt, krankheitsbedingt, mitgezittert, mitzittern, mitzitternden, nachzubauen, notbedingt, pandemiebedingt, plattgebügelten, plattgetrampelt, redimensioniert, sanktionsbedingten, schiefgelaufenen, schiefgelaufener, schiefstehende, schöngeredet, schönreden, schönzureden, sonnenzugewandten, straßenzugewandten, überdimensioniert, unterdimensioniert, vollgesprüht, vollgestapelt, vorbeiflanieren, weltzugewandter, zukunftszugewandte*

Words from section 3.2 *Bekleidungsberuf – Berufsbekleidung, Stiefelleder – Lederstiefel, Fettbauch – Bauchfett, Zugluft – Luftzug, Stallkühe – Kuhstall, Drahtmaschen – Maschendraht, Teppichwolle – Wollteppich, Dauerprojekte – Projektdauer, Ölolíven – Olivenöl, Schalenobst – Obstschale, Schachtelpappe – Pappschachtel, Tütenpapier – Papiertüten, Absatzschuhe – Schuhabsatz, Stoffschichten – Schichtstoffe, Druckkunst – Kunstdrucken*

Words from section 3.3: *Energieentlastungspakets, Energieentlastungspakete, Energieentlastungspaketen, Entlastungspaket, Gasengpässen, Gasengpasses, Gasengpass, Gasengpässe, Mindestfüllstände, Mindestfüllstand, Mindestfüllständen, Mindestfüllstands, Halbleiterengpässe, Halbleiterengpass, Halbleiterengpasses, Halbleiterengpässen, Testmüdigkeit, Testmüdigkeit, Endlos-Lockdown, Distanzlernens, Distanzlernen, distanzlernende, Distanzlernenden, ansteckungsfrei, ansteckungsfreien, ansteckungsfreiem, ansteckungsfreies, bemaskt, Impfbereitschaft, impfbereit, Impffrust, Herzschrittmacherträger, Parkraumbewirtschaftungskonzept, Fluggastdatensätze, Herzschrittmachertypen, Musiktauschbörse, Kochbuchautorinnen, Kochbuchautor, Kochbuchautoren, Atomkraftgegner, Kopfsteinpflasterpassage, massenvernichtungswaffenfreien, Hausstaubmilbenallergikern, Gebärdensprachdolmetscher:innen, Gebärdensprachdolmetscher, Kopftuchträgerin, Schilddrüsenhormontabletten, Sonnenblumenkernöl, Haifischflossensuppe, Abwasserbeseitigungspflicht, Knochenmarkspenderregister, Muttermilchersatzprodukten, Kinderbuchautorin, Maiglöckchenduft, Herrenarmbanduhr, Kunstrasenspielfeld, Kunstrasenspielfeldes, Dornröschendasein, Gabelstaplerführerschein, Festnetztelefonnummer, Massentierhaltungsanlagen, Mauerblümchendasein, Obstbaumschnittkurs, Kreuzworträtselfrage, Medizinjournalismus, Blutzuckerteststreifen, Kuhmilcheiweißallergie, Kleinkläranlage, Kläranlagenbetreiber, Kleinkläranlagenbetreiber*

event_chaos event_happened event_situation event_surprise	The tiger ate the fruit . **It** resulted in chaos . The wolf ate the apple . **It** actually happened . The lion ate the carrot . **It** was a funny situation . The owl ate the cake . **It** came as a surprise .
gender_step	I saw a **pineapple** . **It** was big .
object_overlap_eatdrinkdrink	The zebra ate the fruit and the **monkey** drank the tea . **It** liked the tea .
object_overlap_eatdrinkeat	The **lion** ate the fruit and the zebra drank the milk . **It** liked fruit .
object_verb_overlap_eatdrinkdrink	The mouse ate the cookie and the **bear** drank the milk . **It** drank the milk quickly .
object_verb_overlap_eatdrinkeat	The **zebra** ate the fruit and the lion drank the water . **It** ate the fruit quickly .
pleo_believe pleo_rain pleo_seem pleo_shame	The lion ate the ice cream . **It** is hard to believe this is true . The lion ate the pizza . **It** was raining . The frog ate the cookie . **It** seemed this was unnecessary . The giraffe ate the cheese . **It** is a shame .
subject_overlap_eatdrinkdrink	The turtle ate the bread and the dog drank the **tea** . The dog liked **it** .
subject_overlap_eatdrinkeat	The dove ate the **fruit** and the zebra drank the tea . The dove liked **it** .
subject_verb_overlap_eatdrinkdrink	The dove ate the apple and the frog drank the **water** . The frog drank **it** quickly .
subject_verb_overlap_eatdrinkeat	The mouse ate the **fruit** and the lion drank the tea . The mouse ate **it** quickly .
verb_overlap_eatdrinkdrink	The zebra ate and the **bear** drank . **It** drank quickly .
verb_overlap_eatdrinkeat	The **zebra** ate and the lion drank . **It** ate a lot .
world_knowledge world_knowledge	The **tiger** ate the ice cream . **It** was happy . The giraffe ate the **steak** . **It** was cooked .

Table 15: Overview of the different categories of reference in ContraCat. The noun that is referred to by the *it* in question, as well as the *it* itself, are marked in bold face. For the categories *event_** and *pleo_**, the *it* does not refer to a noun.

Robust MT evaluation with Sentence-level Multilingual Augmentation

Duarte M. Alves*[1], Ricardo Rei[1,3,4], Ana C. Farinha[3],
José G. C. de Souza[3], André F. T. Martins[1,2,3]
[1]Instituto Superior Técnico, University of Lisbon, Portugal
[2]Instituto de Telecomunicações, Lisbon, Portugal
[3]Unbabel, Lisbon, Portugal, [4]INESC-ID, Lisbon, Portugal

Abstract

Automatic translations with critical errors may lead to misinterpretations and pose several risks for the user. As such, it is important that Machine Translation Evaluation systems are robust to these errors in order to increase the reliability and safety of the translation process. Here we introduce SMAUG, a novel Sentence-level Multilingual AUGmentation approach for generating translations with critical errors and apply this approach to create a test set to evaluate the robustness of Machine Translation metrics to these errors. We show that current State-of-the-Art methods are improving their capability to distinguish translations with and without critical errors and to penalize the first accordingly. We also show that metrics tend to struggle with errors related to named entities and numbers and that there is a high variance in the robustness of current methods to translations with critical errors.

1 Introduction

In recent years, Machine Translation (MT) systems have been used in diverse real world environments. However, widespread adoption of these systems raises many concerns, namely in the quality of their outputs. Ideally, human translators would evaluate generated translations but this process is expensive and slow. As an alternative, automatic Machine Translation Evaluation relies on external systems to measure the quality of generated translations.

As a crucial aspect of Machine Translation Evaluation, it is vital to ensure that generated sentences do not contain critical errors. As detailed in Specia et al. (2021), translations with such errors deviate in meaning from their source sentence in ways that may lead to misinterpretations and pose health, safety, legal, reputation, religious or financial implications. Specia et al. (2021) group these translations into three categories, based on how their meaning deviates from the source sentence. Mistranslation errors have critical content in the source sentence translated into a different meaning, not translated (the content remains in the source language), or translated into gibberish. Hallucination errors introduce content in the translated sentence that is not present in the source sentence. Deletion errors exclude important content from the source sentence.

In this work, we propose SMAUG[1], a Sentence-level Multilingual AUGmentation framework to generate translations with critical errors, targeting all the aforementioned critical error categories.

We also introduce a novel test set to analyse the robustness of MT Evaluation systems to critical errors. This test set was created with the proposed augmentation framework and submitted to the WMT22 Challenge Sets Sub-task (Freitag et al., 2022).

Finally, we present the results obtained from evaluating metrics submitted to the WMT22 Metrics Shared Task with the developed test set. We show progress of submitted metrics with respect to baseline systems, particularly concerning Quality Estimation systems. Namely, we demonstrate that several metrics are able to correctly distinguish translations with and without critical errors and to penalize the former. Furthermore, we show that current metrics are less sensitive to translations containing errors in named entities and numbers and that there is a high variance in the performance of current SOTA evaluation metrics with respect to identifying and penalizing the occurrence of critical errors.

2 Related Work

Metrics for Machine Translation Evaluation produce a quality score for a given hypothesis, based on the source sentence and a possibly empty set

*Corresponding author: duartemalves@tecnico.ulisboa.pt

[1] Code available at: https://github.com/Unbabel/smaug

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 469 - 478
December 7-8, 2022 ©2022 Association for Computational Linguistics

of reference translations. These metrics can be divided into two main groups, given their reference set. Reference based metrics have a non-empty reference set, while reference free metrics have an empty reference set. Reference free evaluation is also denominated by Quality Estimation.

Within reference-based metrics, *n*-gram based metrics, such as BLEU (Papineni et al., 2002) and CHRF (Popović, 2015), measure lexical overlap from the hypothesis to the human references. Rei et al. (2020) advocate that these methods fail to capture semantic similarities beyond the lexical level. Their inability to capture meaning at a sentence level also makes them unfit for the detection of critical errors as they equally penalize the usage of synonyms or the mistranslation of a named entity.

As an alternative to *n*-gram matching, more recent methods leverage word representations to capture semantic similarities beyond the lexical level. As described in Rei et al. (2020), *embedding*-similarity methods, like YISI-1 (Lo, 2019) and BERTSCORE (Zhang et al., 2020), create an alignment between the vector representations of the words in the hypothesis and the reference and then compute a score that captures the semantic similarity between both sentences. As noted by Rei et al. (2020), the main issue with these approaches is that human judgements consider other information beyond semantic similarity, limiting the correlation of these methods with human evaluations.

More recently, learnt methods, such as BLEURT20 (Sellam et al., 2020) and COMET (Rei et al., 2020), address this issue by training to directly maximize correlation with human judgements. Results from the WMT21 Metrics (Freitag et al., 2021) and the WMT21 Quality Estimation (Specia et al., 2021) shared tasks suggest that these methods obtain higher correlations with human judgements, such as Direct Assessments (Graham et al., 2013), Human Translation Edit Rate (HTER) (Snover et al., 2006) or Multi-dimensional Quality Metrics (MQM) (Lommel et al., 2014).

However, as noted by Ribeiro et al. (2020), relying on accuracy on held-out sets can lead to an overestimation on the performance of NLP models. As such, Ribeiro et al. (2020) proposes `CheckList` that relies on data augmentation techniques to create examples that test specific behaviours of NLP systems in various situations. Within the field of Machine Translation Evaluation, as a case study for exploring the sensitivity of learnt metrics to specific phenomena, Amrhein and Sennrich (2022) employed Minimum Bayes Risk decoding with COMET as an utility function to identify good hypotheses. The authors show that hypotheses chosen with COMET are more likely to have errors in Named Entities and Numbers when compared to CHRF, indicating the metric is not sensitive enough to these errors.

Considering multiple metrics, Freitag et al. (2021) tested multiple systems on a challenge set with errors related to negation and sentiment polarity and found that most metrics struggle with these errors. Nonetheless, these examples were chosen from existing MT outputs, which can lead to a major human effort, as these errors are not common.

Regarding reference free evaluation, Kanojia et al. (2021) define multiple perturbations to test the robustness of QE systems in detecting specific errors. The authors show that overall the tested perturbations are well detected but some, such as polarity based perturbations, still pose a challenge to QE systems. However, the list of perturbations is not exhaustive and most rely on transformations that do not necessarily preserve the semantics of the phrases, such as random insertions, substitutions and deletions.

3 SMAUG Framework

In order to create an example of a critical error, the proposed framework receives an existing sentence and perturbs it, inducing one of the linguistic phenomena detailed in the following sections. For each linguistic phenomenon, the perturbation process is separated into two phases: transformation and validation. The first phase generates a candidate sentence by perturbing the original translation. This phase may not produce a candidate, as some perturbations are not applicable to all sentences. The second phase verifies whether the produced candidate meets a set of desirable criteria, discarding it otherwise.

3.1 Deviation in Named Entities

The first perturbation replaces a named entity in the original sentence for a different one that is consistent with the original context. The transformation phase of this perturbation, in Figure 1, starts by detecting all Named Entities in the original sentence with the Named Entity Recognition (NER) System in the Stanza library (Qi et al., 2020). If no entity is detected, the generation process stops. Otherwise, a single one is randomly chosen using an Uniform Distribution. This entity is replaced by employing the mT5 pretrained language model (Xue et al., 2021). For this, the span with the sampled entity is replaced by a single mask token and the model is used to generate the candidate sentence. The decoding strategy for the mT5 model is sampling considering the top 50 elements. When compared with other strategies, such as Beam-Search and Top-P sampling, this approach was empirically found to give realistic examples at a lower computational cost. The mT5 model was chosen for three main reasons: it is multilingual and trained on a massive set of different languages; it can generate multiple words from a single mask token, thus not requiring any special strategy for adding mask tokens in order to avoid only single word entities; and does not change the remainder of the sentence, avoiding unwanted side-effects. Nevertheless, the mT5 model was found to often generate punctuation symbols in the beginning of the sentence. In order to increase the credibility of the generated sentences, these symbols were removed.

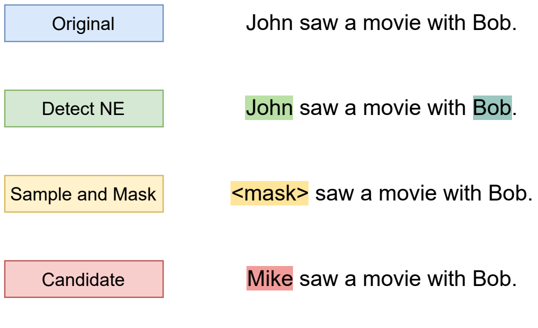

Original
John saw a movie with Bob.
Detect NE
John saw a movie with Bob.
Sample and Mask
<mask> saw a movie with Bob.
Candidate
Mike saw a movie with Bob.

Figure 1: Example of the transformation phase for the Deviation in Named Entities phenomenon.

The validation phase for this perturbation encompasses several sub-validations. On the one hand, in order to ensure the mT5 model generates a named entity, the candidate is only accepted if the above NER model detects the same number of entities in both the candidate and the original sentence. On the other hand, the mT5 model can "guess" the correct named entity from the remaining context. As such, the generated sentence can not be equal to the original. Furthermore, to prevent cases where mT5 produces a small variation of the original entity (for example by adding an hyphen between two words or changing the accentuation), candidate sentences may only be accepted if they have a character-level minimum edit distance to the original above a threshold. This procedure can discard many valid candidates and thus, depending on the desired quality and quantity of generated sentences, may be applied or not. Through manual experimentation, a distance greater or equal to 5 was found to produce a good balance between ensuring the generated entities are different without discarding too many valid candidates. Finally, to increase the overall quality of the generated sentences, several sub-validations can be employed. Candidates with words matching the regular expression of the mT5 masking token (`<extra_id_\d{1,2}>`) are discarded, as they represent cases where the model was unable to generate content. This can be extended by considering more generic expressions such as `extra_*`. Furthermore, since named entities do not usually have characters such as `()[]\{\}_`, candidates that have more of these characters than the original can also be removed. As before, these validations can remove valid candidates and they should be adapted to the use case in question.

3.2 Deviation in Numbers

Another perturbation, similar to the deviation in named entities, replaces a number in the original sentence by a different one. The transformation phase for this phenomenon follows the same procedure as the deviation in Named Entities. However, it employs the regular expression `[-+]?\.?(\d+[.,])*\d+` to detect numbers in the original sentence. From the detected numbers, the process to sample a single number and replace it with another one using the mT5 model is the one described above, from masking the span with the chosen number to generating the candidate sentence.

Regarding the validation phase, it also employs a set of sub-validations. As before, the candidate is accepted only if the regular expression to detect numbers is matched the same number of times in both the original and candidate sentences, ensuring a number was generated. Furthermore, the original and candidate sentences must be different to ensure the mT5 model did not "guess" the number by the context. In this perturbation, the minimum edit distance sub-validation was not applied as small variations in numbers mostly lead to critical errors (for example changing the place of a comma within the number). Finally, candidates matching the mT5 masking token (`<extra_id_\d{1,2}>`) or that introduce one of the following characters

( ) [ ] \ { \ } _ are also removed to increase the overall quality of the generated sentences.

3.3 Deviation in Meaning

Concerning deviations in meaning, a phenomenon that either introduces or removes a negation in the original sentence was developed, thus generating a sentence with the opposite meaning.

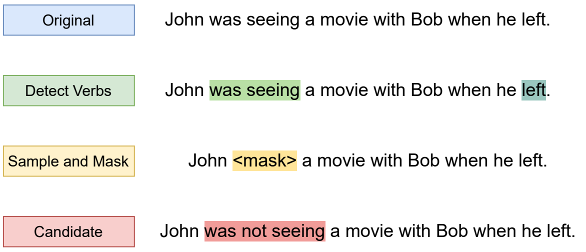
In order to negate the original sentences, this perturbation relies on the POLYJUICE (Wu et al., 2021) model conditioned for negation. POLYJUICE can either negate an entire sentence or a span, by masking the sentence or only the desired text span, respectively. Initial experiments showed that, when trying to negate the entire sentence, the model often forgot some content, specially in longer phrases. Thus, the developed approach, shown in Figure 2, masks a verb in the original sentence, as well as any adjacent auxiliary verbs before it, in order to produce a small perturbation that changes the meaning of the sentence. Specifically, the transform used a Part-of-Speech tagger from the Stanza library (Qi et al., 2020) in the original sentence and recovered all spans with 0 or more AUX tags immediately followed by a VERB tag. If no spans are detected, the generation process stops. Otherwise, one span is sampled using an uniform distribution. Finally, the conditioned POLYJUICE model produces the candidate sentence by negating the original sentence with a mask over the chosen span.

The validation phase for this phenomenon first verifies whether the candidate sentence is equal to the original or if the POLYJUICE model produced its empty token, meaning it was unable to generate a sentence. Furthermore, a RoBERTa (Liu et al., 2019) model trained for Multi-Genre Natural Language Inference (MNLI) corpus was used to verify whether the candidate contradicts the original sentence. This procedure is employed as a proxy for validating whether the generated sentence is a negation over the original.

3.4 Insertion of Content

Regarding the generation of Hallucinated content, a phenomenon to insert new content in the original sentence was devised.

The transformation phase of this perturbation employs a similar strategy to the Named Entities phenomenon. In this case, the masking pattern randomly inserts mask tokens between adjacent words in the original sentence. In order to avoid inserting too much content, a maximum of three mask tokens are introduced. After this step, the masked sentence is fed to the mT5 model, which generates the candidate sentence.



Original
John was seeing a movie with Bob when he left.
Detect Verbs
John was seeing a movie with Bob when he left.
Sample and Mask
John <mask> a movie with Bob when he left.
Candidate
John was not seeing a movie with Bob when he left.

Figure 2: Example of the transformation phase for the Deviation in Meaning phenomenon. Although not shown in this example, the POLYJUICE model receives additional information besides the masked sentence to know the text that was replaced by the mask.

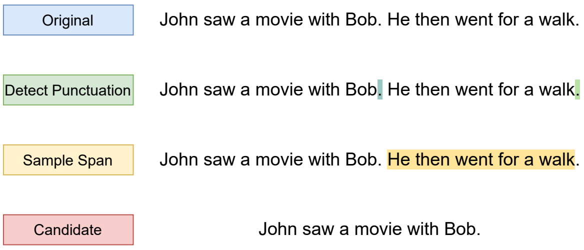
In the validation phase, as in the Named Entities phenomenon, candidate sentences that are equal to the original or that match the regular expression for the mT5 masking pattern (<extra_id_\d{1,2}>) are discarded. Moreover, another sub-validation that ensures the minimum edit distance at a word-level between the candidate and original sentences is above a threshold was applied. As there are only insertions, this sub-validation ensures at least a minimum number of words are introduced in the candidate sentence. Furthermore, higher thresholds increase the likelihood of the candidate sentences having hallucinated content as, with a fixed number of masks (defined in the masking strategy), the model has to generate spans of text with multiple words and it is unlikely that only function words are introduced. Through manual experimentation, a threshold of eight words was found to produce a good balance between ensuring content was added without discarding too many valid candidates.

3.5 Removal of Content

Finally, translations with deletion errors were tackled by a phenomenon that removes a span of text between two punctuation symbols. By considering text spans between adjacent punctuation symbols, this method aims to remove a sub-phrase of the original sentence that likely contains some information, thus generating a sentence which is missing content.

As shown in Figure 3, the transformation phase of this perturbation starts by detecting all instances of the symbols . , ? ! in the original sentence. Then, a span between two adjacent symbols is randomly sampled with an Uniform Distribution. The chosen span, as well as the punctuation symbol after it, are deleted in order to generate the candidate sentence. In order to increase the likelihood of removing content, the deleted span has a minimum number of words. Furthermore, to increase the credibility of the generated sentence, three additional constraints were enforced. First, the first text span was not considered, as translation models are less likely to forget content in the beginning of the sentence. Second, the deleted span has a maximum size, as it is unlikely the translation model drops a large portion of the sentence. Third, if the generated candidate does not end in . ! ? , the final symbol is replaced by a punctuation mark. If no span exists in the previous conditions, the transform does not generate a candidate sentence.

This transform does not require any extra validation, as all verifications are enforced when choosing the text span to delete.



John saw a movie with Bob. He then went for a walk.

John saw a movie with Bob. He then went for a walk.

John saw a movie with Bob. He then went for a walk.

John saw a movie with Bob.

Figure 3: Example of the transformation phase for the Removal of Content phenomenon.

4 Challenge Set

The created test set comprises of records in the format $(s, h_{good}, h_{bad}, r, p)$, where s is a source sentence, h_{good} and h_{bad} are "good" and "bad" hypothesis, r is a reference and p is an identifier for the linguistic phenomenon present in h_{bad}. Three language pairs were considered: English-Portuguese, Spanish-English, Portuguese-English. For each language pair, a data augmentation approach was applied to an existing parallel corpus to generate a the final set of records.

4.1 Parallel Corpora

To create our challenge set we extracted sentences from OPUS (Tiedemann, 2012) ranging several domains such as News and Euro Parliament. To guarantee high-quality references we used Bicleaner tool (Ramírez-Sánchez et al., 2020) with a threshold of 0.85.

4.2 Augmentation Approach

For each language pair, the source side of the respective corpus was considered as source sentences and the target as references. First, the source sentences were translated using an OPUS-MT bilingual model (Tiedemann and Thottingal, 2020)[2]. Second, all the perturbations were applied to the references, generating sentences with at most one critical error. This information was aggregated to create records in the format $(s, h_{good}, h_{bad}, r, p)$, where h_{good} is the translation of the source sentence, h_{bad} is a perturbation of the reference and p is the linguistic phenomenon that was induced. With this approach, multiple records can be created from an original source and reference pair, one for each perturbation applied to the reference. In this case, all the records have the same good hypothesis.

The generated records were then manually filtered and validated to ensure its quality. In this process, we ensured that both the references and the good translations were high quality and that the bad translation contained a critical error. Furthermore, we chose records where h_{good} was different from r to force the metrics to attend to the meaning of the sentence instead of analysing lexical overlap. In the end, around 50 records for each phenomenon and language pair were obtained, as shown in Table 1. The Deviation in Named Entities and Meaning phenomena for the English-Portuguese language pair have 0 records since the Portuguese language is not supported by the NER model in the Stanza library or the POLYJUICE model.

5 Experiments

The developed test set was submitted to the WMT22 Challenge Set Sub-task and the scores for several State-of-the-Art metrics were gathered. The following sections detail the evaluation method for the tested metrics and the obtained results.

5.1 Evaluation Method

We rely on two evaluation methods to assess the robustness of metrics to the developed critical errors.

The first is the official evaluation method for the Shared Task in order to compare the performance of the several metrics. This method used a Kendall-Tau like formulation, defined as:

$$\tau = \frac{Concordant - Discordant}{Concordant + Discordant}, \qquad (1)$$

[2] Available at Hugging Face Transformers (Wolf et al., 2020)

en-pt		pt-en		es-en	
Phenomenon	**Size**	**Phenomenon**	**Size**	**Phenomenon**	**Size**
NE	0	NE	50	NE	48
NUM	49	NUM	48	NUM	50
MEAN	0	MEAN	50	MEAN	48
INS	44	INS	48	INS	50
DEL	48	DEL	50	DEL	49

Table 1: Number of selected records for each phenomenon and language pair. The Deviation in Named Entities and Meaning phenomenon have 0 records for the English-Portuguese language pair as the phenomenon do not support to-Portuguese language pairs.

where $Concordant$ is the number of times the metric assigned a higher score to the good hypothesis and $Discordant$ is the number of times the metric assigned a higher score to the bad hypothesis.

The second method measures the average difference between the scores assigned to h_{good} and h_{bad}, when the score assigned to the h_{good} is higher. For a given set S with pairs of scores, this method is defined as

$$d = \frac{\sum\limits_{(s_{good}, s_{bad}) \in S} \mathbb{I}[s_{good} > s_{bad}](s_{good} - s_{bad})}{\sum\limits_{(s_{good}, s_{bad}) \in S} \mathbb{I}[s_{good} > s_{bad}]} \quad (2)$$

where s_{good} and s_{bad} are respectively the scores for multiple good and bad hypothesis pairs. This formulation is used as a proxy for the confidence of the evaluated metric when it assigns a higher score to the good hypothesis. In order to compare multiple metrics with different scoring intervals, the metric scores are normalized before this evaluation method is applied.

5.2 Baseline Metrics

All the baseline metrics from the Sub-task were considered. These comprise of several State-of-the-Art methods: BLEU and CHRF are *n-gram* based metrics; BERTSCORE and YiSi-1 are *embedding*-similarity methods, and BLEURT20, COMET-20 and COMET-QE are learnt methods.

Figure 4 shows the obtained results for these metrics. For each phenomenon, results show the average Kendall-Tau considering all language pairs and the black bars represent the standard deviation. We observe that the metrics obtain mostly negative correlations, indicating they are assigning higher scores to the bad hypothesis. *n-gram* based metrics show the worst correlations. This result is to be expected as the perturbations create localized changes, such as changing a number, which do not significantly modify the alignments with the reference. Embedding-similarity based metrics exhibit a better performance as contextual embeddings can capture divergence in meaning of the bad hypothesis, but still the obtained correlations are mostly negative. Pretrained models obtain the best results, having positive correlations for the phenomenon Deviation in Meaning, Insertion and Removal of Content. Nevertheless, they still show poor correlations and struggle with Deviation in Named Entities and Numbers.

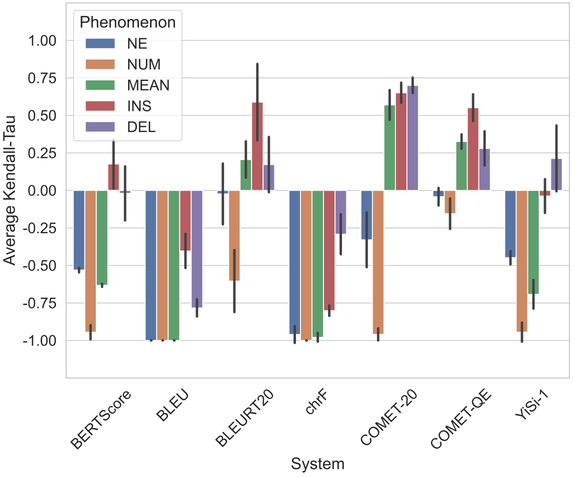


Figure 4: Average Kendall-Tau for baseline metrics discriminated by phenomenon. The coloured bars indicate the average score for all language pairs and the black bars represent the standard deviation.

5.3 Submitted Metrics

The submissions that rely on the reference to predict a score encompass COMET-22 (Rei et al., 2022), metricx_xl_DA_2019[3], MS-COMET-22 (Kocmi et al., 2022) and UniTE (Wan et al., 2022).

[3] Citation was not available.

As depicted in Figure 5, these metrics obtain much higher correlations, when compared to the baselines. The metric metricx_xl_DA_2019 obtains the overall best results, achieving high correlations for all phenomena. Across all metrics, the Deviation in Numbers phenomenon is the one with lowest scores. Furthermore, it is also the one with the highest standard deviation over the several language pairs, showing the uncertainty of these metrics when faced with this perturbation.

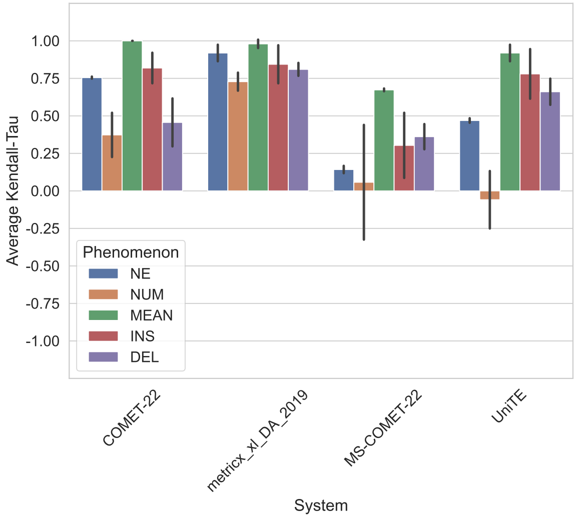


Figure 5: Average Kendall-Tau for submitted reference based metrics discriminated by phenomenon. The coloured bars indicate the average score for all language pairs and the black bars represent the standard deviation.

Regarding reference free metrics, submissions comprise of COMET-Kiwi (Rei et al., 2022), HWTSC-Teacher-Sim (Liu et al., 2022), HWTSC-TLM (Liu et al., 2022), KG-BERTScore (Liu et al., 2022) and MS-COMET-QE-22 (Kocmi et al., 2022). Here, it is important to note that HWTSC-TLM is a system that only receives the hypothesis as input and, as such, it is likely in disadvantage in this task, as the developed bad hypothesis are only critical errors in the context of the source sentence.

As shown in Figure 6, several reference free metrics obtain very high correlations for all linguistic phenomena. The main exception is HWTSC-TLM, which can be attributed to the reasons explained above. KG-BERTScore obtains the best overall results, with almost perfect correlations. Furthermore, we observe that reference free metrics outperform reference based metrics. This result is further discussed in the following section.

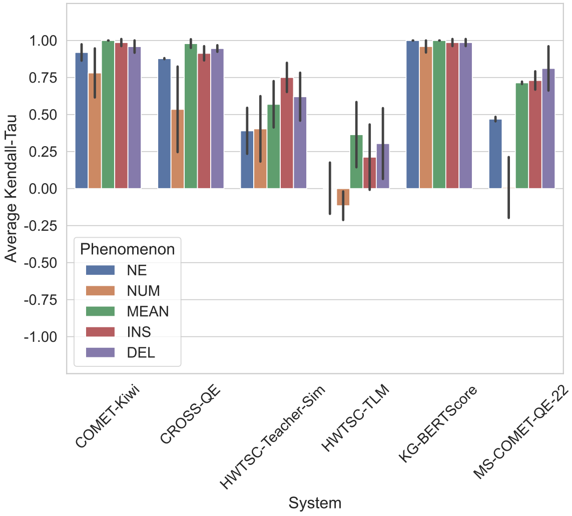


Figure 6: Average Kendall-Tau for submitted reference free metrics discriminated by phenomenon. The coloured bars indicate the average score for all language pairs and the black bars represent the standard deviation.

5.4 Reference based vs Reference free

Figure 7 compares the performance of reference based and reference free metrics across all phenomena. We observe that reference free metrics obtain higher correlations on all perturbations, which can be attributed to the adversarial nature of the bad hypothesis that is specifically generated with a localized perturbation of the reference.

This result reveals the dependency of reference based metrics on the reference and, in particular, on the word overlap of the reference with the hypothesis. Reference-free metrics are forced to attend to the source and compare its meaning with the hypothesis, as there is little word overlap between the two sentences. This issue is particularly visible in the Deviation in Named Entities and Numbers phenomena, where the reference and bad hypothesis differ on a single named entity or number, respectively.

Comparing the performance of metrics for each phenomenon, we verify that both groups of metrics obtain lower correlations for Deviation in Named Entities and Numbers, indicating these phenomena are not well detected by current methods. Moreover, the results show large standard deviations, suggesting an inherent unpredictability on the performance of current methods for all phenomena.

5.5 Penalisation of critical errors

In order to measure whether the metrics penalize the critical errors when they score the bad hypoth-

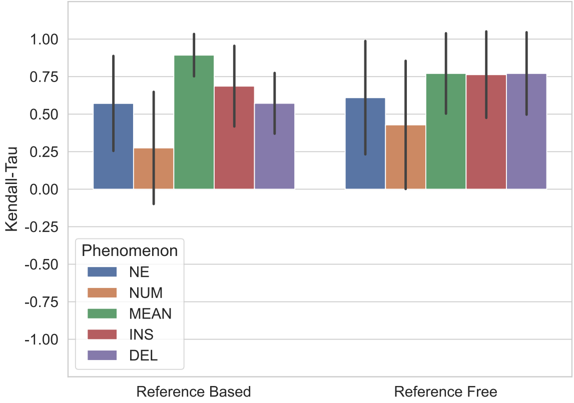


Figure 7: Average Kendall-Tau for submitted reference based and reference free metrics discriminated by phenomenon. The coloured bars indicate the average score for all language pairs and the black bars represent the standard deviation.

esis lower, we compare their Kendall-Tau values with their average difference between the scores for good and bad hypothesis, as described in Section 5.1.

In Figure 8, we observe that submitted metrics not only obtain higher correlations but also have a greater difference between the scores attributed to the good and bad hypothesis. Moreover, the two variables follow a linear relationship, obtaining a Pearson Correlation Coefficient of 0.8924. This shows the metrics that correctly distinguish the good from the bad hypothesis also penalize the bad hypothesis accordingly.

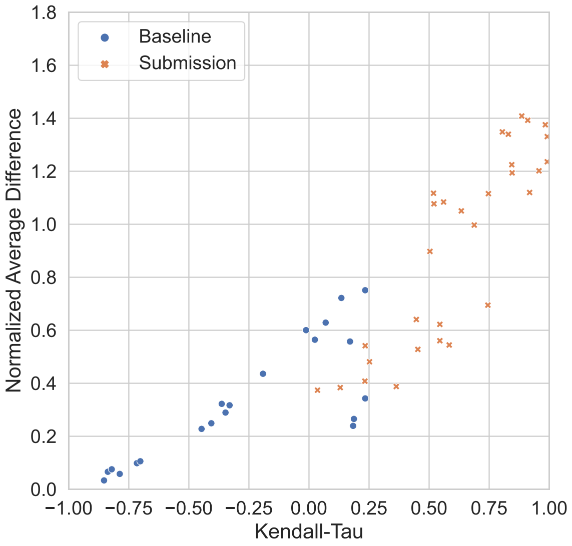


Figure 8: Average Kendall-Tau and Difference for all metrics. Each data point represents a single metric and language pair.

6 Conclusions

Ensuring generated translations do not have critical errors is a crucial aspect of Machine Translation Evaluation, as they can pose various risks. In this work, we propose SMAUG, a multilingual augmentation framework to create translations with critical errors by inducing several linguistic phenomena in existing translations. We also apply these perturbations to create a manually verified test set to assess the robustness of Machine Translation Evaluation systems to critical errors.

With the created test set, we evaluate multiple metrics and show promising progress in current State-of-the-Art methods in both distinguishing translations with and without critical errors and significantly penalizing the occurrence of critical errors in translations. Nevertheless, errors related to named entities and numbers were found to pose a challenge for several tested metrics. Additionally, we observe a high variance in the measured correlations across all the developed phenomena, suggesting an unpredictability on the performance of current methods with respect to detecting critical errors.

One of the challenges in the automatic generation of translations with critical errors is the validation of the output. In this work, we relied on a preliminary automatic validation but also required a manual verification of the outputs. Future work will explore high-precision validation techniques, such as the work of Raunak et al. (2022) that uses very specific detectors to find examples of critical errors in translations.

Furthermore, support for multiple languages is a crucial aspect of this framework. However, several of the devised perturbations support a limited number of languages pairs. For example, the Deviation in Meaning phenomenon only supports to-English language pairs, as the POLYJUICE model is an English only model. A future avenue of research will investigate methods to expanding the number of languages supported by the linguistic phenomena.

Acknowledgements

This work was supported by the P2020 programs (MAIA, contract 045909), by EU's Horizon Europe Research and Innovation Actions (UTTER, contract 101070631), by the European Research Council (ERC StG DeepSPIN 758969), and by the Fundação para a Ciência e Tecnologia through contract UIDB/50008/2020.

References

Chantal Amrhein and Rico Sennrich. 2022. Identifying Weaknesses in Machine Translation Metrics Through Minimum Bayes Risk Decoding: A Case Study for COMET. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, Online. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. Results of WMT22 Metrics Shared Task: Stop Using BLEU – Neural Metrics Are Better and More Robust. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Diptesh Kanojia, Marina Fomicheva, Tharindu Ranasinghe, Frédéric Blain, Constantin Orăsan, and Lucia Specia. 2021. Pushing the right buttons: Adversarial evaluation of quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 625–638, Online. Association for Computational Linguistics.

Tom Kocmi, Hitokazu Matsushita, and Christian Federmann. 2022. MS-COMET: Larger Filtered Human Annotations Help Metric Performance. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Yilun Liu, Xiaosong Qiao, Zhanglin Wu, Su Chang, Min Zhang, Yanqing Zhao, Song Peng, shimin tao, Hao Yang, Ying Qin, Jiaxin Guo, Minghan Wang, Yinglu Li, Peng Li, and Xiaofeng Zhao. 2022. Partial could be better than whole. HW-TSC 2022 Submission for the Metrics Shared Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Arle Lommel, Aljoscha Burchardt, and Hans Uszkoreit. 2014. Multidimensional Quality Metrics (MQM): A Framework for Declaring and Describing Translation Quality Metrics. *Tradumàtica: tecnologies de la traducció*, 0:455–463.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón, and Sergio Ortiz Rojas. 2020. Bifixer and bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298, Lisboa, Portugal. European Association for Machine Translation.

Vikas Raunak, Matt Post, and Arul Menezes. 2022. SALTED: A Framework for SAlient Long-Tail Translation Error Detection.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 Submission for the Metrics Shared Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4902–4912, Online. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the WMT 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 684–725, Online. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Yu Wan, Dayiheng Liu, Baosong Yang, Haibo Zhang, Boxing Chen, Derek Wong, and Lidia Chao. 2022. UniTE: Unified translation evaluation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8117–8127, Dublin, Ireland. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6707–6723, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *8th International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia. OpenReview.net.

ACES: Translation Accuracy Challenge Sets for Evaluating Machine Translation Metrics

Chantal Amrhein[1*] and **Nikita Moghe**[2*] and **Liane Guillou**[2*]
[1]Department of Computational Linguistics, University of Zurich
[2]School of Informatics, University of Edinburgh
amrhein@cl.uzh.ch, nikita.moghe@ed.ac.uk, lguillou@ed.ac.uk

Abstract

As machine translation (MT) metrics improve their correlation with human judgement every year, it is crucial to understand the limitations of such metrics at the segment level. Specifically, it is important to investigate metric behaviour when facing accuracy errors in MT because these can have dangerous consequences in certain contexts (*e.g.,* legal, medical). We curate ACES[1], a Translation **A**ccuracy **C**halleng**E** **S**et, consisting of 68 phenomena ranging from simple perturbations at the word/character level to more complex errors based on discourse and real-world knowledge. We use ACES to evaluate a wide range of MT metrics including the submissions to the WMT 2022 metrics shared task and perform several analyses leading to general recommendations for metric developers. We recommend: a) combining metrics with different strengths, b) developing metrics that give more weight to the source and less to surface-level overlap with the reference and c) explicitly modelling additional language-specific information beyond what is available via multilingual embeddings.

1 Introduction

Challenge sets have already been created for measuring the success of systems or metrics on a particular phenomenon of interest for a range of NLP tasks, including but not limited to: Sentiment Analysis[2] (Li et al., 2017; Mahler et al., 2017; Staliūnaitė and Bonfil, 2017), Natural Language Inference (McCoy and Linzen, 2019; Rocchietti et al., 2021), Question Answering (Ravichander et al., 2021), Machine Reading Comprehension (Khashabi et al., 2018), Machine Translation (MT) (King and Falkedal, 1990; Isabelle et al., 2017), and the more specific task of pronoun translation in MT (Guillou and Hardmeier, 2016). They are useful to compare the performance of different systems, or to identify performance improvement/degradation between a modified system and a previous iteration.

In this work, we describe the University of Zurich - University of Edinburgh submission to the *Challenge Sets* subtask of the Conference on Machine Translation (WMT) 2022 Metrics shared task. Our Translation **A**ccuracy **C**halleng**E** **S**et (ACES) consists of 36,476 examples covering 146 language pairs and representing challenges from 68 phenomena (see Appendix A.4 for the distribution of examples across language pairs and Appendix A.5 for the distribution of language pairs across phenomena). We focus on translation accuracy errors and base the phenomena covered in our challenge set on the Multidimensional Quality Metrics (MQM) ontology (Lommel et al., 2014). We include phenomena ranging from simple perturbations involving the omission/addition of characters or tokens, to more complex examples involving mistranslation e.g. ambiguity and hallucinations in translation, untranslated elements of a sentence, discourse-level phenomena, and real-world knowledge. We evaluate the metrics submitted to the WMT 2022 metrics shared task and a range of baseline metrics on ACES. Additionally, we perform an extensive analysis, which aims to reveal:

1. The extent to which reference-based and reference-free metrics take into account the source sentence context.

2. The extent to which reference-based metrics rely on surface-level overlap with the reference.

3. Whether using multilingual embeddings results in better metrics.

[*] Equal contribution by all authors.

[1] Our dataset is available at https://huggingface.co/datasets/nikitam/ACES and the corresponding evaluation scripts at https://github.com/EdinburghNLP/ACES

[2] Submitted to the EMNLP 2017 "Build It Break It" shared task on sentiment analysis

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 479 - 513
December 7-8, 2022 ©2022 Association for Computational Linguistics

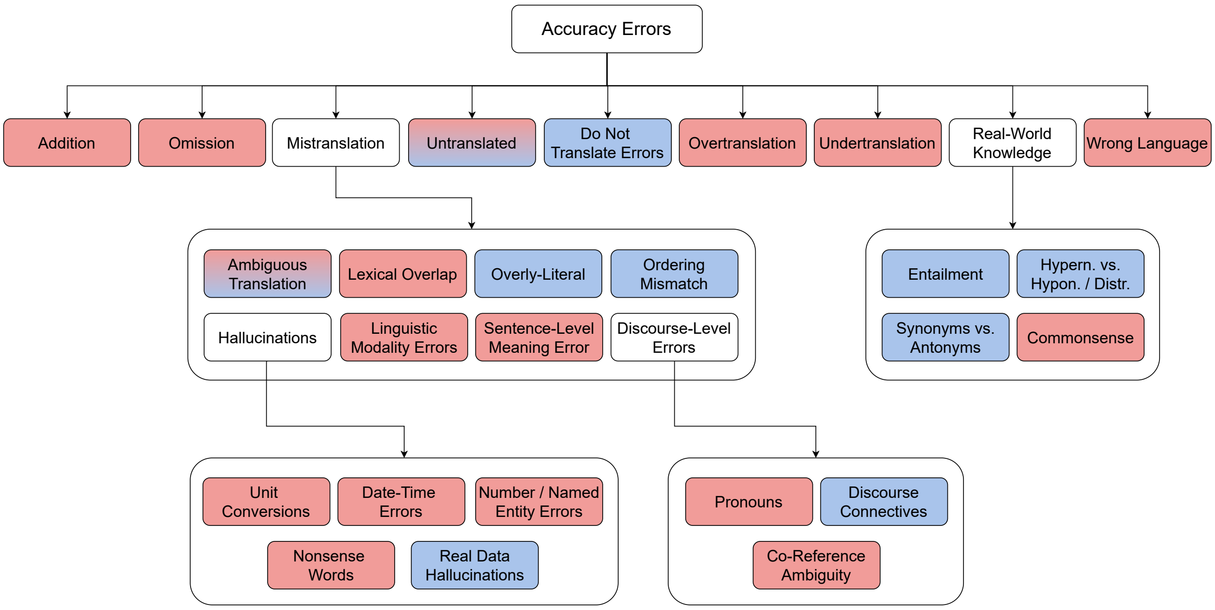


Figure 1: Diagram of the error categories on which our collection of challenge sets is based. Red means challenge sets are created automatically, blue means challenge sets are created manually.

Based on our analysis, we recommend that metric developers consider: a) combining metrics with different strengths, e.g. in the form of ensemble models, b) paying more attention to the source and avoiding reliance on surface-overlap with the reference, and c) explicitly modelling additional language-specific information beyond what is available via multilingual embeddings. We also propose that ACES be used as a benchmark for developing evaluation metrics for MT to monitor which error categories can be identified better, and also whether there are any categories for which metric performance degrades.

2 Motivation

With the advent of neural networks and especially Transformer-based architectures (Vaswani et al., 2017), machine translation outputs have become more and more fluent (Bentivogli et al., 2016; Toral and Sánchez-Cartagena, 2017; Castilho et al., 2017). Fluency errors are also judged less severely than accuracy errors by human evaluators (Freitag et al., 2021a) which reflects the fact that accuracy errors can have dangerous consequences in certain contexts, for example in the medical and legal domains (Vieira et al., 2021).

For these reasons, we decided to build a challenge set focused on accuracy errors. Specifically, we use the hierarchy of errors under the class *Accuracy* from the MQM ontology to design these challenge sets. We extend this ontology by two error classes (translations defying real-world knowledge and translations in the wrong language) and specify several more specific subclasses such as discourse-level errors or ordering mismatches. A full overview of all error classes can be seen in Figure 1. Our challenge set consists of synthetically generated adversarial examples, examples from repurposed contrastive MT test sets (both marked in red), and manually annotated examples (marked in blue). To create the challenge sets, we use test sets from tasks such as adversarial paraphrase detection, Natural Language Inference, and contrastive MT test sets created independently of the WMT shared tasks to avoid overlap with the data that is used to train neural evaluation metrics.

Another aspect we focus on is including a broad range of language pairs in ACES. Whenever possible we create examples for all language pairs covered in a source dataset when we use automatic approaches. For phenomena where we create examples manually, we also aim to cover at least two language pairs per phenomenon, but are of course limited to the languages spoken by the authors.

Finally, we aim to offer a collection of challenge sets covering both easy and hard phenomena. While it may be of interest to the community to continuously test on harder examples to check where machine translation evaluation metrics still break, we believe that easy challenge sets are just as important to ensure that metrics do not suddenly become worse at identifying error types that were previously considered “solved”. Therefore, we take a holistic view when creating ACES and do not filter out individual examples or exclude challenge sets based on baseline metric performance or other factors.

We first discuss previous efforts to create challenge sets (Section 3), before giving a broad overview of the datasets used to construct ACES (Section 4) and discussing the individual challenge sets in more detail (Section 5). We then introduce the metrics that participated in the shared task (Section 6), present an overview of their performance on ACES (Section 7) and detailed analyses (Section 8) that lead to a set of recommendations for future metric development (Section 9).

3 Related Work

Challenge sets are used to study a particular phenomenon of interest rather than the general distribution of phenomena in standard test sets (Popović and Castilho, 2019). The earliest introduction of challenge sets was by King and Falkedal (1990) who probed acceptability of machine translations for different domains. Challenge sets have been prevalent in different fields within NLP such as parsing (Rimell et al., 2009), NLI (McCoy and Linzen, 2019; Rocchietti et al., 2021), question answering (Ravichander et al., 2021), reading comprehension (Khashabi et al., 2018) and sentiment analysis (Li et al., 2017; Mahler et al., 2017; Staliūnaitė and Bonfil, 2017), to name a few. These challenge sets provide insights on whether state-of-the-art models are robust to domain shifts, and whether they have some understanding of linguistic phenomena like negation/commonsense or they simply rely on shallow heuristics. Another line of work under “adversarial datasets” also focuses on creating examples by perturbing the standard test set to fool the model (Smith (2012); Jia and Liang (2017), *inter-alia*).

Challenge sets for evaluating MT systems have focused on the translation models’ ability to generate the correct translation given a phenomenon of interest. These include word sense ambiguity (Vamvas and Sennrich, 2021), gender bias (Rudinger et al., 2017; Zhao et al., 2018; Stanovsky et al., 2019), structural divergence (Isabelle et al., 2017) and discourse level phenomena (Guillou and Hardmeier, 2016; Emelin and Sennrich, 2021).

While such challenge sets focus on evaluating specific machine translation models, it is necessary to identify whether the existing machine translation evaluation metrics also perform well under these and related phenomena. Developing challenge sets for machine translation metric evaluation has gained considerable interest because recently, neural MT evaluation metrics have shown improved correlation with human judgements (Freitag et al., 2021b; Kocmi et al., 2021). However, their weaknesses remain relatively unknown and only a small number of works (e.g. Hanna and Bojar (2021) and Amrhein and Sennrich (2022)) have proposed systematic analyses to uncover them.

Previous challenge sets for metric evaluation focused on negation and sentiment polarity (Specia et al., 2020) and synthetic perturbations such as antonym replacement, word omission, number swapping, punctuation removal, etc. (Freitag et al., 2021b). Avramidis et al. (2018) developed a manually constructed test suite of linguistically motivated perturbations for identifying weaknesses in reference-free evaluation. However, these challenge sets for metrics are only focused on high-resource language pairs such as English↔German and English→Chinese. In this work, we repurpose existing machine translation challenge sets to evaluate machine translation evaluation metrics. We introduce several synthetically generated and manually created challenge sets that broadly focus on translation accuracy errors for 146 language pairs.

4 Datasets

The majority of the examples in our challenge set were based on data extracted from three main datasets: FLORES-101, PAWS-X, and XNLI (with additional translations from XTREME).

The **FLORES-101** evaluation benchmark (Goyal et al., 2022) consists of 3,001 sentences extracted from English Wikipedia and translated into 101 languages by professional translators. **FLORES-200** (NLLB Team et al., 2022) expands the set of languages in FLORES-101. Originally intended for multilingual and low-resource MT evaluation, these datasets have a particular focus on low-resource languages.

PAWS-X (Yang et al., 2019), a cross-lingual dataset for paraphrase identification, consists of pairs of sentences that are labelled as true or adversarial paraphrases. It comprises the Wikipedia portion of the PAWS corpus (Zhang et al., 2019) translated from English into six languages: French, Spanish, German, Chinese, Japanese, and Korean.

The development and test sets (23,659 sentences total) were manually translated by professional translators, and the training set was translated using NMT systems via Google Cloud Translation[3].

XNLI (Conneau et al., 2018) is a multilingual Natural Language Inference (NLI) dataset consisting of 7,500 premise-hypothesis pairs with their corresponding inference label. The English examples were generated by crowd source workers before being manually translated into 14 languages: French, Spanish, German, Greek, Bulgarian, Russian, Turkish, Arabic, Vietnamese, Thai, Chinese, Hindi, Swahili and Urdu. In addition, we use the automatic translations from **XTREME** (Hu et al., 2020) of the XNLI test set examples from these 14 languages into English.

For the mistranslation phenomena Gender in Occupation Names and Word Sense Disambiguation, we leveraged the WinoMT and MuCoW datasets. **WinoMT** (Stanovsky et al., 2019), a challenge set developed for analysing gender bias in MT, contains 3,888 English examples extracted from the Winogender (Rudinger et al., 2017) and WinoBias (Zhao et al., 2018) coreference test sets. WinoMT sentences cast participants into non-stereotypical gender roles and the dataset has an equal balance of male and female genders, and of stereotypical and non-stereotypical gender-role assignments (e.g., a female nurse vs. a female doctor). **MuCoW** (Raganato et al., 2019) is a multilingual contrastive, word sense disambiguation test suite for machine translation. The dataset covers 16 language pairs with more than 200,000 contrastive sentence pairs. It was automatically constructed from word-aligned parallel corpora and BabelNet's (Navigli and Ponzetto, 2012) wide-coverage multilingual sense inventory.

For the discourse-level phenomena, we relied on *annotated* resources developed specifically to support work on those phenomena in an MT setting. The **WMT 2018 English-German pronoun translation evaluation test suite** (Guillou et al., 2018) contains 200 examples of the ambiguous English pronouns *it* and *they* extracted from the TED talks portion of ParCorFull (Lapshinova-Koltunski et al., 2018). The example sentences were translated into German by the 16 English-German systems submitted to WMT 2018, and the (German) pronoun translations were manually judged by human annotators as "good/bad". **Wino-X** (Emelin and Sennrich, 2021) is a parallel dataset of German, French, and Russian Winograd schemas, aligned with their English counterparts. It was developed for commonsense reasoning and coreference resolution and used for this purpose to generate examples for Commonsense Co-Reference Disambiguation. The **Europarl ConcoDisco** corpus (Laali and Kosseim, 2017) comprises the English-French parallel texts from Europarl (Koehn, 2005) over which automatic methods were used to perform PDTB-style discourse connective annotation. Discourse connectives are labelled with their sense type and are aligned between the two languages.

5 Challenge Sets

Creating a contrastive challenge set for evaluating a machine translation evaluation metric requires a source sentence, a reference translation, and two translation hypotheses: one which contains an error or phenomenon of interest (the "incorrect" translation) and one which is a correct translation in that respect (the "good" translation). One possible way to create such challenge sets is to start with two alternative references (or two identical copies of the same reference) and insert errors into one of them to form an incorrect translation while the uncorrupted version can be used as the good translation. This limits the full evaluation scope to translation hypotheses that only contain a single error. To create a more realistic setup, we also create many challenge sets where the good translation is not free of errors, but it is a better translation than the incorrect translation. For automatically created challenge sets, we put measures in place to ensure that the incorrect translation is indeed a worse translation than the good translation.

5.1 Addition and Omission

We create a challenge set for addition and omission errors which are defined in the MQM ontology as "target content that includes content not present in the source" and "errors where content is missing from the translation that is present in the source", respectively. We focus on the level of constituents and use an implementation by Vamvas and Sennrich (2022) to create synthetic examples of addition and omission errors.

To generate examples, we use the concatenated dev and devtest sets from the FLORES-101 evaluation benchmark. We focus on the 46 languages

[3] https://cloud.google.com/translate

for which there exists a stanza parser[4] and create datasets for all languages paired with English plus ten additional language pairs that we selected randomly. The script by Vamvas and Sennrich (2022) randomly drops constituents from the source sentence and then generates two translations, one of the full source and one of the partial source without the constituent. Here is an example of two resulting translations:

Full:	For example, castle visits in the Loire Valley, the Rhine Valley, or a cruise **to interesting cities on the Danube** or **a** boat ride along the Erie Canal.
Partial:	For example, castle visits in the Loire Valley, the Rhine Valley, or a cruise or boat ride along the Erie Canal.

Only partial translations that can be constructed by deleting spans from the full translation are considered. For translation, we use the M2M100[5] model with 1.2B parameters (Fan et al., 2021).

We create **omission** examples by taking the original source and reference and using the translation of the full source as a good translation and the translation of the partial source as an incorrect translation. For **addition** errors, we test if the deleted span also occurs in the reference. If it doesn't, we discard the example, if it does, we delete that span from the reference and pair this partial reference with the partial source. Then, the good translation is the translation of the partial source and the incorrect translation is the translation of the full source. For language pairs with a BLEU score of less than 13 between the good translation and the reference, we manually check the examples to ensure the challenge set features appropriate examples of additions and omissions.

5.2 Mistranslation - Ambiguous Translation

This error type is defined in the MQM ontology as a case where "an unambiguous source text is translated ambiguously". For this error type, we create challenge sets where MT metrics are presented with an unambiguous source and an ambiguous reference. The metrics then need to choose between two disambiguated translation hypotheses where only one meaning matches the source sentence. Therefore, these challenge sets test whether metrics consider the source when the reference is not expressive enough to identify the better translation. Since many reference-based metrics, by design, do not include the source to compute evaluation scores, we believe that this presents a challenging test set.

Our method for creating examples is inspired by Vamvas and Sennrich (2021) who score a translation against two versions of the source sentence, one with an added correct disambiguation cue and one with a wrong disambiguation cue to determine whether a translation model produced the correct translation or not. Instead of adding the disambiguation cues to the source, we use an unambiguous source and add disambiguation cues to an ambiguous reference to create two contrasting translation hypotheses.

5.2.1 Ambiguity - Occupation Names Gender

First, we create a challenge set based on WinoMT, where the challenge is to choose either a translation with a "female" or "male" disambiguation cue based on the source sentence:

SRC (de):	Der Manager feuerte **die** Bäcker**in**.
REF (en):	The manager fired the baker.
✓:	The manager fired the **female** baker.
✗:	The manager fired the **male** baker.

We take all English sentences from the WinoMT dataset where either a pro-stereotypical or an anti-stereotypical occupation name occurs. The original sentences in WinoMT contain additional context from which the gender in the English sentence can be inferred. For example, the sentence above exists in the dataset once as "The manager fired the baker because she was too rebellious." from which it is clear that the baker is female, and once as "The manager fired the baker because he was upset." from which it is clear that the manager is male. To make the English sentences ambiguous, we remove the explanatory subordinate clauses using a sequence of regular expressions, so that the sentence becomes "The manager fired the baker." where the gender of the manager and the baker are ambiguous.

We then add the disambiguation cues ("female" or "male") to the ambiguous English sentences and translate them into German, French and Italian which are all languages that mark gender morphologically on most nouns that refer to a person. For translation, we use Google Translate[6] because we find that this system produces gendered occupation

[4] https://stanfordnlp.github.io/stanza/available_models.html

[5] https://huggingface.co/facebook/m2m100_1.2B

[6] https://translate.google.com/

names that are largely faithful to the disambiguation cues. Finally, we remove explicit translations of "female" and "male" from the German, French or Italian output that would help the disambiguation beyond morphological cues. We predict the gender of the occupation names using the scripts provided by Stanovsky et al. (2019) and only keep translation pairs where both the translation of the male-disambiguated source is predicted to be male and the translation of the female-disambiguated source is predicted to be female. We then use either the German, French or Italian translation as the source sentence, the disambiguated English sentences as the translation candidates, and the ambiguous English sentence as the reference, as shown in the example above.

5.2.2 Ambiguity - Word Sense Disambiguation

Second, we create a challenge set based on MuCoW, where the challenge is to choose a translation with a sense-matching disambiguation cue based on the unambiguous source sentence:

SRC (de):	Was heisst "**Brühe**"?
REF (en):	What does "**stock**" mean?
✓:	What does "**vegetable stock**" mean?
✗:	What does "**penny stock**" mean?

We start with disambiguation cues that were automatically extracted by Vamvas and Sennrich (2021) via masked language modelling. Initial screening of the data shows that some disambiguation cues are not sense-specific enough. Therefore, we decide to manually check all disambiguation cues and ensure they are sense-specific and if necessary, replace them with other cues. We generate three pairs of contrasting disambiguation cues per example and use the question "What does X mean?" as a pattern to create the challenge set examples. We decided against using sentences where ambiguous words occur naturally since it may be possible to infer the correct sense from the context of the English sentence rather than by looking at the unambiguous source word. We annotate each example as to whether the correct sense is the more frequent or less frequent sense using frequency counts provided by Vamvas and Sennrich (2021). Following this methodology, we create challenge sets for German into English and Russian into English.

5.2.3 Ambiguity - Discourse Connectives

Third, we create a challenge set where the challenge is to identify a translation with the correct discourse connective based on the unambiguous source sentence:

SRC (fr):	Aucun test de qualité de l'air n'ait été réalisé dans ce bâtiment **depuis** notre élection.
REF (en):	No air quality test has been done on this particular building **since** we were elected.
✓:	No air quality test has been done on this particular building **from the time** we were elected.
✗:	No air quality test has been done on this particular building **because** we were elected.

The English discourse connective "since" can have either causal or temporal meaning, which is expressed explicitly in both French and German. Exploiting this fact, we use the ambiguous "since" in the reference and create two contrastive translations one with "because" for causal meaning and one with "from the time" for temporal meaning. The correct translation is determined by looking at the French or German source sentence where this information is marked explicitly. We use the discourse connective annotations in the Europarl ConcoDisco corpus for this challenge set. We use an automatic-guided search based on the French discourse connective "depuis" (which has temporal meaning) to identify candidate translation pairs. We then manually construct valid contrasting examples for causal and temporal "since" based on the English reference. This results in a challenge set for French-English but we also create a German-English version of the challenge set, where we translate the French source sentences into German and manually correct them.

5.3 Mistranslation - Hallucinations

In this category, we group together several subcategories of mistranslation errors that happen at the word level and could occur due to hallucination by an MT model. Such errors are wrong units, wrong dates or times, wrong numbers or named entities, as well as hallucinations at the subword level that result in nonsensical words. We also present a challenge set of annotated hallucinations in real MT outputs. These challenge sets test whether the machine translation evaluation metrics can reliably identify hallucinations when presented with a correct alternative translation.

5.3.1 Hallucination - Date-Time Errors

We create a challenge set for the category of “date-time errors”. To do this, we collect month names and their abbreviations for several language pairs. We then form a good translation by swapping a month’s name with its abbreviation. The corresponding incorrect translation is generated by swapping the month name with another month name:

SRC (pt): Os manifestantes esperam coletar uma petição de 1,2 milhão de assinaturas para apresentar ao Congresso Nacional em **novembro**.

REF (en): Protesters hope to collect a petition of 1.2 million signatures to present to the National Congress in **November**.

✓: The protesters expect to collect a petition of 1.2 million signatures to be submitted to the National Congress in **Nov.**

✗: The protesters expect to collect a petition of 1.2 million signatures to be submitted to the National Congress in **August**.

To create this dataset, we use the automatic translations of the FLORES-101 dataset from Section 5.1. We choose all pairs with target languages for which we know the abbreviations for months[7] which results in 70 language pairs. As a measure of control, we check that the identified month names in the translation also occur in the reference. If they do not, we exclude the example.

5.3.2 Hallucination - Numbers and Named Entities

We create a challenge set for numbers and named entities where the challenge is to identify translations with incorrect numbers or named entities. Following the analysis by Amrhein and Sennrich (2022), we perform character-level edits (adding, removing or substituting digits in numbers or characters in named entities) as well as word-level edits (substituting whole numbers or named entities). In the 2021 WMT metrics shared task, number differences were not a big issue for most neural metrics (Freitag et al., 2021b). However, we believe that simply changing a number in an alternative translation and using this as an incorrect translation as done by Freitag et al. (2021b) is an overly simplistic setup and does not cover the whole translation hypothesis space.

To address this shortcoming, we propose a three-level evaluation (see examples below). The first, easiest level follows Freitag et al. (2021b) and applies a change to an alternative translation to form an incorrect translation. The second level uses an alternative translation that is lexically very similar to the reference as the good translation and applies a change to the reference to form an incorrect translation. The third, and hardest level, uses an alternative translation that is lexically very different from the reference as the good translation and applies a change to the reference to form an incorrect translation. In this way, our challenge set tests whether number and named entity differences can still be detected as the surface similarity between the two translation candidates decreases and the surface similarity between the incorrect translation and the reference increases.

SRC (es): Sin embargo, Michael Jackson, Prince y **Madonna** fueron influencias para el álbum.

REF (en): Michael Jackson, Prince and **Madonna** were, however, influences on the album.

Level-1 ✓: However, Michael Jackson, Prince, and **Madonna** were influences on the album.

Level-1 ✗: However, Michael Jackson, Prince, and **Garza** were influences on the album.

Level-2 ✓: However, Michael Jackson, Prince, and **Madonna** were influences on the album.

Level-2 ✗: Michael Jackson, Prince and **Garza** were, however, influences on the album.

Level-3 ✓: The record was influenced by **Madonna**, Prince, and Michael Jackson though.

Level-3 ✗: Michael Jackson, Prince and **Garza** were, however, influences on the album.

We use cross-lingual paraphrases from the PAWS-X dataset as a pool of alternative translations to create this challenge set. For levels 2 and 3, we measure surface-level similarity with Levenshtein distance[8] at the character-level and use spacy[9] (Honnibal et al., 2020) for identifying named entities of type “person”. To substitute whole named entities, we make use of the `names`[10] Python library. We only consider language pairs for which we can use a spacy NER model on the target side, which results in 42 language pairs.

[7] `https://web.library.yale.edu/cataloging/months`

[8] `https://github.com/life4/textdistance`

[9] `https://spacy.io/`

[10] `https://github.com/treyhunner/names`

5.3.3 Hallucination - Unit Conversion

We create a challenge set for unit conversions where the challenge is to identify the correct unit conversion:

SRC (de):	Auf einem **100 Fuß** langen Teilabschnitt läuft Wasser über den Damm.
REF (en):	Water is spilling over the levee in a section **100 feet** wide.
✓:	On a **30.5 metres** long section, water flows over the dam.
✗:	On a **100 metres** long section, water flows over the dam.

We take all source sentences, reference sentences and translations of the FLORES-101 sets from Section 5.1. We only use the 45 language pairs into English since the Python packages we use for unit conversion only work for English. We first use the Python package `quantulum3`[11] to extract unit mentions from text. We only consider sentences where we identify the same unit mentions in the translation as in the reference and we remove self-disambiguating unit mentions, like "645 miles (1040 km)" from the reference and translation. Then, we use the Python package `pint`[12] to convert unit mentions in the translation into different units. The permitted conversions are listed in Appendix A.2.

The sentence with the converted amount and new unit is considered to be the good translation. Based on this sentence, we construct two incorrect versions, one where the amount matches the reference but the unit is still converted (see example above) and one where the amount is the converted amount but the unit is copied from the reference. We pair each incorrect translation with the good translation and add both examples to the challenge set individually. We are aware that this challenge set lies beyond the ability of current MT systems and evaluation metrics, however, we believe challenge sets such as these incentivise future work on such capabilities which would reduce the workload in post-editing.

5.3.4 Hallucination - Nonsense Words

We also consider more natural hallucinations at the subword level. Because recent MT systems are trained with subwords (Sennrich et al., 2016), an MT model may choose a wrong subword at a specific time step such that the resulting token is not a known word in the target language. With this challenge set, we are interested in how well neural MT evaluation metrics that incorporate subword-level tokenisation can identify such "nonsense" words.

To create this challenge set, we consider tokens which are broken down into at least two subwords and then randomly swap those subwords with other subwords to create nonsense words. In the example below, "mass" is broken down as "mas" and "##s" using subwords and the new word is created by swapping "mas" with "in" while retaining "##s", creating "ins" as the nonsense word. We use the paraphrases from the PAWS-X dataset as good translations and randomly swap one subword in the reference to generate an incorrect translation. This perturbation is language-agnostic. We use the multilingual BERT (Devlin et al., 2019) tokeniser to replace the subwords.

SRC (de):	Die **Massen**produktion von elektronischen und digitalen Filmen war bis zum Aufkommen der pornographischen Videotechnik direkt mit der Mainstream-Filmindustrie verbunden.
REF (en):	The **mas**s production of electronic and digital films was directly linked to the mainstream film industry until the emergence of pornographic video technology.
✓:	Until the advent of pornographic video technology , the mass production of electronic and digital films was tied directly to the mainstream film industry.
✗:	The **in**s production of electronic and digital films was directly linked to the mainstream film industry until the emergence of pornographic video technology.

5.3.5 Hallucination - Real Data Hallucinations

The previously discussed hallucination challenge sets were all created automatically. In addition to these challenge sets, we also create one with real data hallucinations.

For this dataset, we manually check the translations of the FLORES-101 dev and devtest sets for four language pairs: de→en, en→de, fr→de and en→mr. We consider both cases where a more frequent, completely wrong word occurs and cases where the MT model started with the correct subword but then produced random subwords as hallucinations. Translations with a hallucination are used as incorrect translations. We manually replace the hallucination part with its correct translation to form the good translation. If possible, we create one good translation by copying the corresponding

[11] https://github.com/nielstron/quantulum3

[12] https://github.com/hgrecco/pint

token(s) from the reference and one with a synonymous token that does not match the reference:

SRC (de): Es wird angenommen, dass dieser voll gefiederte warmblütige Raubvogel aufrecht auf zwei Beinen lief und **Krallen** wie der Velociraptor hatte.

REF (en): This fully feathered, warm blooded bird of prey was believed to have walked upright on two legs with **claws** like the Velociraptor.

✓ (copy): It is believed that this fully feathered warm-blooded predator ran upright on two legs and had **claws** like the Velociraptor.

✓ (syn.): It is believed that this fully feathered warm-blooded predator ran upright on two legs and had **talons** like the Velociraptor.

✗: It is believed that this fully feathered warm-blooded predator ran upright on two legs and had **crumbs** like the Velociraptor.

5.4 Mistranslation - Lexical Overlap

Language models trained with the masked language modelling objective are successful on downstream tasks because they model higher-order word co-occurrence statistics instead of syntactic structures (Sinha et al., 2021). Although this has been shown for a monolingual English model, we expect that multilingual pre-trained models, as well as MT metrics finetuned on such models, exhibit such behaviour. Similarly, existing surface-level metrics rely on n-gram matching between the hypothesis and the reference. Thus, we are interested in whether MT evaluation metrics can reliably identify the incorrect translation if it shares a high degree of lexical overlap with the reference:

SRC (fr): En 1924, il a été porte-parole invité de l'ICM à Toronto, à Oslo en 1932 et à Zurich en 1936.

REF (en): In 1924 he was an invited spokesman for the ICM in Toronto, in **Oslo in 1932** and in **1936 in Zurich.**

✓: He served as a guest speaker for ICM in 1924, 1932 and 1936 in Toronto, Oslo and Zurich.

✗: He was an invited spokesman for the ICM in Toronto in 1924, in **Zurich in 1932** and in **Oslo in 1936.**

In this example, Oslo and Zurich are swapped in the "incorrect translation" making the sentence factually incorrect. To create such examples, we use the PAWS-X dataset for which adversarial paraphrase examples were constructed by changing the word order and/or the syntactic structure while maintaining a high degree of lexical overlap. We only consider examples in the development set that are adversarial paraphrases.

We automatically translate the first example in a pair (fr→en, en→fr, en→ja) and then manually correct the translations for en, fr, and ja to obtain 100 "good translations" per language. We use the corresponding first paraphrase as the "reference" and the second (adversarial) paraphrase as the "incorrect translation". We then pair these examples with the first paraphrase in the remaining six languages in PAWS-X to obtain the "source". Following this methodology we create examples for each target language (xx→en, xx→fr, xx→ja).

5.5 Mistranslation - Linguistic Modality

Modal auxiliary verbs signal the function of the main verb that they govern. For example, they may be used to denote possibility ("could"), permission ("may"), the giving of advice ("should"), or necessity ("must"). We are interested in whether MT evaluation metrics can identify when modal auxiliary verbs are incorrectly translated:

SRC (de): Mit der Einführung dieser Regelung **könnte** diese Freiheit enden.

REF (en): With this arrangement in place, this freedom **might** end.

✓: With the introduction of this regulation, this freedom **could** end.

✗: With the introduction of this regulation, this freedom **will** end.

We focus on the English modal auxiliary verbs: "must" (necessity), and "may", "might", "could" (possibility). We begin by identifying parallel sentences where there is a modal verb in the German source sentence and one from our list (above) in the English reference. We then translate the source sentence using Google Translate to obtain the "good" translation and manually replace the modal verb with an alternative with the same meaning where necessary (e.g. "have to" denotes necessity as does "must"; also "might", "may" and "could" are considered equivalent). For the incorrect translation, we manually substitute the modal verb that conveys a different meaning or *epistemic strength* e.g. in the example above "might" (possibility) is replaced with "will", which denotes (near) certainty. Instances of "may" with *deontic* meaning (e.g. expressing permission) are excluded from the set, leaving only those with an *epistemic* meaning (expressing probability or prediction). We also construct examples in which the modal verb is omitted from the incorrect translation.

We employ two strategies to create examples: one in which the modal auxiliary is substituted, and another where it is deleted. We use a combination of the FLORES-200 and PAWS-X datasets as the basis of the challenge sets.

5.6 Mistranslation - Overly Literal Translations

MQM defines this error type as translations that are overly literal, for example literal translations of figurative language. Here, we look specifically at idioms and at real-data errors.

5.6.1 Overly Literal - Idioms

Idioms tend to be translated overly literally (Dankers et al., 2022) and it is interesting to see if such translations are also preferred by neural machine translation evaluation metrics, which likely have not seen many idioms during finetuning:

SRC (de):	Er hat versucht, mir die Spielregeln zu erklären, aber **ich verstand nur Bahnhof**.
REF (en):	He tried to explain the rules of the game to me, but **I did not understand them**.
✓:	He tried to explain the rules of the game to me, but **it was all Greek to me**.
✗:	He tried to explain the rules of the game to me, but **I only understood train station**.

We create this challenge set based on the PIE[13] parallel corpus of English idiomatic expressions and literal paraphrases (Zhou et al., 2021). We manually translate 102 parallel sentences into German for which we find a matching idiom that is not a word-by-word translation of the original English idiom. Further, we create an overly-literal translation of the English and German idioms. We use either the German or English original idiom as the source sentence. Then, we either use the correct idiom in the other language as the reference and the literal paraphrase as the good translation, or vice versa. The incorrect translation is always the overly-literal translation of the source idiom.

5.6.2 Overly-Literal - Real Data Errors

We are also interested in overly-literal translations occurring in real data:

SRC (de):	Today, the only insects that cannot fold back their wings are **dragon flies** and mayflies.
REF (en):	Heute sind **Libellen** und Eintagsfliegen die einzigen Insekten, die ihre Flügel nicht zurückklappen können.
✓ (copy) :	Heute sind die einzigen Insekten, die ihre Flügel nicht zurückbrechen können, **Libellen** und Mayflies.
✓ (syn.):	Heute sind die einzigen Insekten, die ihre Flügel nicht zurückbrechen können, **Wasserjungfern** und Mayflies.
✗:	Heute sind die einzigen Insekten, die ihre Flügel nicht zurückbrechen können, **Drachenfliegen** und Mayflies.

For this challenge set, we manually check MT translations of the FLORES-101 datasets. If we find an overly-literal translation, we manually correct it to form the good translation. We create one good translation where we copy the part of the reference that corresponds to the overly-literal part and, if possible, another good translation where we use a synonym of the reference token. This challenge set contains examples for four language pairs: de→en, en→de, fr→de and en→mr.

5.6.3 Mistranslation - Sentence-Level Meaning Error

We also consider a special case of sentence-level semantic error that arises due to the nature of the task of Natural Language Inference (NLI). The task of NLI requires identifying where the given hypothesis is an entailment, contradiction, or neutral, with respect to a given premise. As a result, the premise and hypothesis have substantial overlap but they vary in meaning. We are interested in whether MT evaluation metrics can pick up on such sentence-level meaning changes:

SRC (el):	Ο πραγματικός ϑόρυβος ελκύει τους ηλικιωμένους.
REF (en):	Real noise appeals to the old. (premise)
✓:	The real noise attracts the elderly.
✗:	Real noise appeals to the young and appalls the old. (hypothesis)

We use the XNLI dataset to create such examples. We consider examples where there is at least 0.5 chrF score between the English premise and hypothesis and where the labels are either contradiction or neutral. Examples with an entailment label are excluded as some examples in the dataset are paraphrases of each other and there would be no sentence-level meaning change. We discuss ef-

[13] https://github.com/zhjjn/MWE_PIE

fects of entailment in Section 5.12.1. We use either the premise or the hypothesis as the reference and an automatic translation as the "good translation". The corresponding premise or hypothesis from the remaining 14 languages is used as the source. The "incorrect translation" is either the premise if the reference is the hypothesis, or vice versa.

5.7 Mistranslation - Ordering Mismatch

We also investigate the effects of changing word order in a way that changes meaning:

SRC (de):	Erfülle Dein Zuhause mit einem köstlichem **Kaffee** am Morgen und etwas entspannendem **Kamillentee** am Abend.
REF (en):	Fill your home with a rich **coffee** in the morning and some relaxing **chamomile tea** at night.
✓:	Fill your home with a delicious **coffee** in the morning and some relaxing **chamomile tea** in the evening.
✗:	Fill your home with a delicious **chamomile tea** in the morning and some relaxing **coffee** in the evening.

This challenge set is created manually by changing translations from the FLORES-101 dataset and covers de→en, en→de and fr→de.

5.8 Mistranslation - Discourse-level Errors

We introduce a new subclass of mistranslation errors that specifically cover discourse-level phenomena.

5.8.1 Discourse-level Errors - Pronouns

First, we are interested in how MT evaluation metrics handle various discourse-level phenomena related to pronouns. To create these challenge sets, we use the English-German pronoun translation evaluation test suite from the WMT 2018 shared task as the basis for our examples.

We extract all translations (by the English-German WMT 2018 systems) that were marked as "correct" by the human annotators, for the following six categories derived from the manually annotated pronoun function and attribute labels: pleonastic *it*, anaphoric subject and non-subject position *it*, anaphoric *they*, singular *they*, and group *it/they*. In the case of anaphoric pronouns, we select only the inter-sentential examples (i.e. where the sentence contains both the pronoun and its antecedent). We use the MT translations as the "good" translations and automatically generate "incorrect" translations using one of the following strategies: *omission* - the translated pronoun is deleted from the MT output, *substitution* - the "correct" pronoun is replaced with an "incorrect" form.

For *anaphoric* pronouns, when translated from English into a language with grammatical gender, such as German, the pronoun translation must a) agree in number and gender with the translation of its antecedent, and b) have the correct grammatical case. We propose "incorrect" translations as those for which this agreement does not hold:

SRC (en):	I have a *shopping bag*; ***it*** is red.
REF (de):	Ich habe eine *Einkaufstüte*; ***sie*** ist rot.
✓:	Ich habe einen *Einkaufsbeutel*; ***er*** ist rot.
✗ (subs.):	Ich habe einen *Einkaufsbeutel*; ***sie*** ist rot.
✗ (omit):	Ich habe einen *Einkaufsbeutel*; ***Ø*** ist rot.

Conversely, for *pleonastic* uses of "it" no agreement is required, instead, the correct translation in German requires a simple mapping: "it" → "es". An ‘incorrect" translation of pleonastic ‘it' in German could be "er" (masc. sg.) or "sie" (fem. sg., or pl.). We create, for each "correct" translation a set of possible "incorrect" values and automatically select one at random to replace the "correct" pronoun. For example, in the pleonastic case:

SRC (en):	**It** is raining
REF (de):	**Es** regnet
✓:	**Es** regnet
✗ (subs.):	**Er** regnet
✗ (omit):	**Ø** regnet

5.8.2 Discourse-level Errors - Discourse Connectives

The English discourse connective "while" is ambiguous – it may be used with either a *Comparison.Contrast* or *Temporal.Synchrony* sense – as are two of its possible translations into French: "tandis que" and "alors que". We leverage a corpus of parallel English/French sentences with discourse connectives marked and annotated for sense, and select examples with ambiguity in the French source sentence. We construct the good translation by replacing instances of "while" temporal with "as" or "as long as" and instances of "while" comparison as "whereas" (ensuring grammaticality is preserved). For the incorrect translation, we replace the discourse connective with one with the alternative sense of "while" e.g. we use "whereas" (comparison) where a temporal sense is required:

SRC (fr):	Dans l'UE-10, elles ont progressé de 8% **tandis que** la dette pour l'UE-2 a augmenté de 152%.
REF (en):	In EU-10 they grew by 8% **while** the debt for the EU-2 increased by 152%.
✓:	In the EU-10, they increased by 8% **when** the debt for the EU-2 increased by 152%.
✗:	In the EU-10, they increased by 8% **whereas** the debt for the EU-2 increased by 152%.

We extract our examples from the Europarl ConcoDisco dataset. We automatically selected the sentence pairs that contain an instance of "while" in English and either "alors que" or "tandis que" in French. Our dataset contains examples for both the *Comparison.Contrast* sense and the *Temporal.Synchrony* sense.

This challenge set complements the discourse connectives set in section 5.2.3, in which the English discourse connective "since" is ambiguous, but the corresponding connectives in French and German are not. Note that while in the previous challenge set the correct translation can be identified by looking at the source, here metrics can only rely on context to identify the correct discourse connective.

5.8.3 Discourse-level Errors - Commonsense Co-Reference Disambiguation

One of the greater challenges within computational coreference resolution is referring to the correct antecedent by using commonsense/real-world knowledge. Emelin and Sennrich (2021) construct a benchmark to test whether multilingual language models and neural machine translation models can perform such commonsense coreference resolutions. We are interested in whether such commonsense coreference resolutions pose a challenge for MT evaluation metrics:

SRC (en):	It took longer to clean the fish tank than the dog cage because **it** was dirtier.
REF (de):	Das Reinigen des Aquariums dauerte länger als das des Hundekäfigs, da **es** schmutziger war.
✓:	Das Reinigen des Aquariums dauerte länger als das des Hundekäfigs, da **das Aquarium** schmutziger war.
✗ :	Die Reinigung des Aquariums dauerte länger als die des Hundekäfigs, da **er** schmutziger war.

The English sentences in the Wino-X challenge set were sampled from the Winograd schema. All contain the pronoun *it* and were manually translated into two contrastive translations for de, fr, and ru. Based on this data, we create our challenge sets covering two types of examples: For the first, the good translation contains the pronoun referring to the correct antecedent, while the incorrect translation contains the pronoun referring to the incorrect antecedent. For the second, the correct translation translates the instance of *it* into the correct disambiguating filler, while the second translation contains the pronoun referring to the incorrect antecedent (see example above).

The sentences for en→de were common across both the challenge sets developed by Emelin and Sennrich (2021). Hence, the corresponding correct translations from the two challenge sets were used as the "good" translation for our evaluation setup. For en→ru and en→fr, the source containing the ambiguous pronoun was machine translated and then verified by human annotators to form the "good" translation.

5.9 Untranslated

MQM defines this error type as "errors occurring when a text segment that was intended for translation is left untranslated in the target content". In ACES, we consider both word-level and sentence-level untranslated content.

5.9.1 Untranslated - Word-Level

For word-level untranslated content, we manually annotate translations of the FLORES-101 dev and devtest sets:

SRC (fr):	À l'origine, l'émission mettait en scène des **comédiens de doublage** amateurs, originaires de l'est du Texas.
REF (de):	Die Sendung hatte ursprünglich lokale Amateur**synchronsprecher** aus Ost-Texas.
✓ (copy):	Ursprünglich spielte die Show mit Amateur**synchronsprechern** aus dem Osten von Texas.
✓ (syn.):	Ursprünglich spielte die Show mit Amateur-**Synchron-Schauspielern** aus dem Osten von Texas.
✗:	Ursprünglich spielte die Show mit Amateur-**Doubling-Schauspielern** aus dem Osten von Texas.

We do not only count complete copies as untranslated content but also content that clearly comes from the source language but was only adapted to look more like the target language (as in the example above). If we encounter an untranslated span, we use this translation as the incorrect translation and create a good translation by copying the correct span from the reference and, if possible, a second good translation where we use a synonym for the correct reference span. We manually annotate such untranslated errors for en→de, fr→de, de→en, en→mr.

5.9.2 Untranslated - Full Sentences

In the case of underperforming machine translation models, sometimes the generated output contains a majority of the tokens from the source language to the extent of copying the entire source sentence.[14] We create a challenge set by simply copying the entire source sentence as the incorrect translation. We used a combination of examples from the FLORES-200, XNLI, and PAWS-X datasets to create these examples.

We expect that this challenge set is likely to break embedding-based, reference-free evaluation because the representation of the source and the incorrect translation will be the same, thus leading to a higher score.

5.10 Do Not Translate Errors

This category of errors is defined in MQM as content in the source that should be copied to the output in the source language, but was mistakenly translated into the target language. Common examples of this error type are company names or slogans. Here, we manually create a challenge set based on the PAWS-X data which contains many song titles that should not be translated:

SRC (en):	Dance was one of the inspirations for the exodus - song **"The Toxic Waltz"**, from their 1989 album "Fabulous Disaster".
REF (de):	Dance war eine der Inspirationen für das Exodus-Lied **„The Toxic Waltz"** von ihrem 1989er Album „Fabulous Disaster".
✓:	Der Tanz war eine der Inspirationen für den Exodus-Song **„The Toxic Waltz"**, von ihrem 1989er Album „Fabulous Disaster".
✗:	Der Tanz war eine der Inspirationen für den Exodus-Song **„Der Toxische Walzer"**, von ihrem 1989er Album „Fabulous Disaster".

To construct the challenge set, we use one paraphrase as the good translation and manually translate an English sequence of tokens (e.g. a song title) into German to form the incorrect translation.

5.11 Overtranslation and Undertranslation

Hallucinations from a translation model can often produce a term which is either more generic than the source word or more specific. Within the MQM ontology, the former is referred to as undertranslation while the latter is referred to as overtranslation. For example, "car" may be substituted with "vehicle" (undertranslation) or "BMW" (overtranslation). To automate the generation of such errors, we use Wordnet (Miller, 1994). In our setup a randomly selected noun from the reference translation is replaced by its corresponding hypernym or hyponym to simulate undertranslation or overtranslation errors, respectively:

SRC (de):	Bob und Ted waren Brüder. Ted ist der **Sohn** von John.
REF (en):	Bob and Ted were brothers. Ted is John's **son**.
✓:	Bob and Ted were brothers, and Ted is John's **son**.
✗:	Bob and Ted were brothers. Ted is John 's **male offspring**.

During the implementation, we only replaced the first sense listed in Wordnet for the corresponding noun, which may not be appropriate in the given translation. We constructed this challenge set for hypernyms and hyponyms using the PAWS-X dataset, only considering the language pairs where the target language is English.

5.12 Real-world Knowledge

We manually constructed examples each for en→de and de→en for the first four phenomena described in this section. We used German-English examples from XNLI, plus English translations from XTREME as the basis for our examples. Typically, we select a single sentence, either the premise or hypothesis from XNLI, and manipulate the MT translations.

5.12.1 Real-world Knowledge - Textual Entailment

We test whether the metrics can recognise textual entailment – that is, whether a metric can recognise that the meaning of the source/reference is entailed by the "good" translation. We construct examples for which the good translation entails the meaning of the original sentence (and its reference). For example, we use the entailment *was murdered* → *died* (i.e. if a person is murdered then they must have died) to construct the good translation in the

[14] Through observations of Swahili → English translation; unpublished work

example above. We construct the incorrect translation by replacing the entailed predicate (*died*) with a related but non-entailed predicate (here *was attacked*) – a person may have been murdered without being attacked, i.e. by being poisoned for example. When constructing our examples we focus solely on leveraging *directional entailments*. We specifically exclude paraphrases as these are bidirectional.

In cases where an antonymous predicate is available, we use that predicate in the incorrect translation. For example, if "lost" is in the source/reference, we use "won" in the incorrect translation (lost $\not\Rightarrow$ won).

SRC (de):	Ein Mann **wurde ermordet**.
REF (en):	A man **was murdered**.
✓:	A man **died**.
✗ (omit):	A man **was attacked**.

5.12.2 Real-world Knowledge - Hypernyms and Hyponyms

We consider a translation that contains a *hypernym* of a word to be better than one that contains a *hyponym*. For example, whilst translating "Hund" ("dog") with the broader term "animal" results in some loss of information, this is preferable over hallucinating information by using a more specific term such as "labrador" (i.e. an instance of the hyponym class "dog"):

SRC (de):	..., dass der **Hund** meiner Schwester gehört.
REF (en):	... the **dog** belonged to my sister.
✓ (hypernym):	... the **pet** belonged to my sister.
✗ (hyponym):	... the **labrador** belonged to my sister.

We used Wordnet and WordRel.com[15] (an online dictionary of words' relations) to identify hypernyms and hyponyms of nouns within the reference sentences, and used these as substitutions in the MT output: hypernyms are used in the "good" translations and hyponyms in the "incorrect" translations.

5.12.3 Real-world Knowledge - Hypernyms and Distractors

Similar to the hypernym vs. hyponym examples, we construct examples in which the good translation contains a hypernym (here "pet") of the word in the reference (here "dog"). We form the incorrect translation by replacing the original word in the source/reference with a different member from the same class (here "cat"; both cats and dogs belong to the class of pets). For example:

SRC (de):	..., dass der **Hund** meiner Schwester gehört.
REF (en):	... the **dog** belonged to my sister.
✓ (hypernym):	... the **pet** belonged to my sister.
✗ (hyponym):	... the **cat** belonged to my sister.

As before, we used Wordnet and WordRel.com to identify hypernyms of nouns present in the reference translation.

5.12.4 Real-world Knowledge - Antonyms

Similar to the generation of over- and undertranslations, we also constructed "incorrect" translations by replacing words with their corresponding antonyms from Wordnet. We construct challenge sets for both nouns and verbs.

For nouns, we automatically constructed "incorrect" translations by replacing nouns in the reference with their antonyms. The "good" translation is not amended. This method may result in noisy replacement of nouns with their respective antonyms.

In the case of verbs, we manually constructed a more challenging set of examples intended to be used to assess whether the metrics are able to distinguish between translations that contain a synonym versus an antonym of a given word. We replaced verbs in the reference with a synonym to produce the good translation, and with their antonym to produce the incorrect translation:

SRC (de):	Ich **hasste** jedes Stück der Schule!
REF (en):	I **hated** every bit of school!
✓ (synonym):	I **loathed** every bit of school!
✗ (antonym):	I **loved** every bit of school!

For the verbs challenge set, we consider a translation that contains a synonym of a word in the reference to be a "good" translation, and one that contains an antonym of that word to be "incorrect". As in the example above the use of synonyms preserves the meaning of the original sentence, and the antonyms introduce a polar opposite meaning.

5.12.5 Real-world Knowledge - Commonsense

We are also interested in whether evaluation metrics prefer translations that adhere to common sense. To test this, we remove explanatory subordinate

[15] https://wordrel.com/

clauses from the sources and references in the dataset described in Section 5.8.3. This guarantees that when choosing between the good and incorrect translation, the metric cannot infer the correct answer from looking at the source or the reference:

SRC (en):	Die Luft im Haus war kühler als in der Wohnung.
REF (de):	The air in the house was cooler than in the apartment.
✓:	The air in the house was cooler than in the apartment because **the apartment** had a broken air conditioner.
✗:	The air in the house was cooler than in the apartment because **the house** had a broken air conditioner.

We remove the explanatory subordinate clauses using a sequence of regular expressions. We then pair the shortened source and reference sentences with the full translation that follows commonsense as the good translation and the full translation with the other noun as the incorrect translation.

Since we present several challenge sets in Section 5.2 where the good translation can only be identified by looking at the source sentence, we also create a version of this challenge set where the explanatory subordinate clause is only removed from the reference but not from the source. By comparing this setup with the results from the setup described above, we achieve another way of quantifying how much a metric considers the source.

5.13 Wrong Language

Most of the representations obtained from large multilingual language models do not explicitly use the language identifier (id) as an input while encoding a sentence. Here, we are interested in checking whether sentences which have similar meanings are closer together in the representation space of neural MT evaluation metrics, irrespective of their language. We create a challenge set for embedding-based metrics where the incorrect translation is in a similar language (same typology/same script) to the reference (e.g. a Catalan translation may be used as the incorrect translation if the target language is Spanish). Note that this is also a common error with multilingual machine translation models. We constructed these examples using the FLORES-200 dataset where the "good" translation was the automatic translation and the "incorrect" translation was the reference from a language similar to the target language:

SRC (en):	Cell comes from the Latin word cella which means small room.
REF (es):	El término célula deriva de la palabra latina cella, que quiere decir «cuarto pequeño».
✓ (es):	La célula viene de la palabra latina cella que significa habitación pequeña.
✗ (ca):	Cèl·lula ve de la paraula llatina cella, que vol dir habitació petita.

We construct two categories within this challenge set: one where the target language is a higher-resource language and the incorrect language is a lower-resource language and vice-versa. The languages we consider are (`src-tgt-sim`): en-hi-mr, en-es-ca, en-cs-pl, fr-mr-hi, en-pl-cs, and en-ca-es.

Note that if we were to compare references for different languages and not an automatic translation vs. a reference, this challenge set should be considered unsolvable for reference-free metrics if there is no way to specify the desired target language. But in this case, we expect reference-free metrics to prefer the reference that we use as the "incorrect translation" since there may be translation errors in the automatically translated "good translation".

5.14 Fluency

Although the focus of ACES is on accuracy errors, we also include a small set of fluency errors for the punctuation category. Future work might consider expanding this set to include other categories of fluency errors.

5.14.1 Punctuation

We assess the effect of deleting and substituting punctuation characters. We employ four strategies: 1) deleting all punctuation, 2) deleting only quotation marks (i.e. removing indications of quoted speech), 3) deleting only commas (i.e. removing clause boundary markers), 4) replacing exclamation points with question marks (i.e. statement → question).

In strategies 1 and, especially, 3 and 4, some of the examples may also contain accuracy-related errors. For example, the meaning of the sentence could be changed in the incorrect translation if we remove a comma, e.g. in the (in)famous example "Let's eat, Grandma!" vs. "Let's eat Grandma!". We use the TED Talks from the WMT 2018 English-German pronoun translation evaluation test suite and apply all deletions and substitutions automatically.

6 Evaluation Methodology

We shall now briefly describe the metrics that participated in the challenge set shared task. The organisers of the shared task also provided scores by a number of baseline metrics, as described below.

6.1 Baseline Metrics

BLEU (Papineni et al., 2002) compares the token-level n-grams of the hypothesis with the reference translation and then computes a precision score weighted by a brevity penalty.

spBLEU (Goyal et al., 2022) is BLEU computed over text tokenised with a single language-agnostic SentencePiece subword model. The spBLEU baselines, F101SPBLEU and F200SPBLEU, are named according to whether the SentencePiece tokeniser (Kudo and Richardson, 2018) was trained using data from the FLORES-101 or FLORES-200 languages.

chrF (Popović, 2017) evaluates translation outputs based on a character n-gram F-score by computing overlaps between the hypothesis and the reference.

BERTScore (Zhang et al., 2020) uses contextual embeddings from pre-trained language models to compute the similarity between the tokens in the reference and the generated translation using cosine similarity. The similarity matrix is used to compute precision, recall, and F1-scores.

BLEURT20 (Sellam et al., 2020) is a BERT-based (Devlin et al., 2019) regression model, which is first trained on scores of automatic metrics/similarity of pairs of reference sentences and their corrupted counterparts. It is then fine-tuned on the WMT human evaluation data to produce a score for a hypothesis given a reference translation.

COMET-20 (Rei et al., 2020) uses a cross-lingual encoder (XLM-R (Conneau et al., 2020)) and pooling operations to obtain sentence-level representations of the source, hypothesis, and reference. These sentence embeddings are combined and then passed through a feedforward network to produce a score. COMET is trained on human evaluation scores of machine translation systems submitted to WMT until 2020.

COMET-QE was trained similarly to COMET-20 but as this is a reference-free metric, only the source and the hypothesis are combined to produce a final score.

YiSi-1 (Lo, 2019) measures the semantic similarity between the hypothesis and the reference by using cosine similarity scores of multilingual representations at the lexical level. It optionally uses a semantic role labeller to obtain structural similarity. Finally, a weighted f-score based on structural and lexical similarity is used for scoring the hypothesis against the reference.

6.2 Metrics Submitted to WMT 2022

We list the descriptions provided by the authors of the respective metrics and refer the reader to the relevant system description papers for further details.

COMET-22 (Rei et al., 2022) is an ensemble between a vanilla COMET model trained with Direct Assessment (DA) scores and a Multitask model that is trained on regression (MQM regression) and sequence tagging (OK/BAD word identification from MQM span annotations). These models are ensembled together using a hyperparameter search that weights different features extracted from these two evaluation models and combines them into a single score. The vanilla COMET model is trained with DA's ranging 2017 to 2020 while the Multitask model is trained using DA's ranging from 2017 to 2020 plus MQM annotations from 2020 (except for en-ru that uses TedTalk annotations from 2021).

Metric-X is a massive multi-task metric, which fine tunes large language model checkpoints such as mT5 on a variety of human feedback data such as Direct Assessment, MQM, QE, NLI and Summarization Eval. Scaling up the metric is the key to unlocking quality and makes the model work in difficult settings such as evaluating without a reference, evaluating short queries, distinguishing high quality outputs, and evaluating on other generation tasks such as summarisation. The four metrics are referred to according to the mT5 model variant used (xl or xxl) and the fine-tuning data: METRICX_*_DA_2019 only used 2015-19 Direct Assessment data for fine-tuning, whereas METRICX_*_MQM_2020 used a mixture of Direct Assessment 2015-19 and MQM 2020 data.

MS-COMET-22 and **MS-COMET-QE-22** (Kocmi et al., 2022) are built on top of the COMET (Rei et al., 2020) architecture. They are trained on a several times larger set of human judgements covering 113 languages and covering 15 domains. Furthermore, the authors propose filtering of human judgements with potentially low quality. MS-COMET-22 receives the source, the MT hypothesis and the human reference as input, while MS-COMET-QE calculates scores in a quality estimation fashion with access only to the source segment and the MT hypothesis.

UniTE (Wan et al., 2022), Unified Translation Evaluation, is a metric approach where the model-based metrics can possess the ability of evaluating translation outputs following all three evaluation scenarios, i.e. source-only, reference-only, and source-reference-combined. These are referred to in this paper as UNITE-SRC, UNITE-REF, and UNITE respectively.

COMET-Kiwi (Rei et al., 2022) ensembles two QE models similarly to COMET-22. The first model follows the classic Predictor-Estimator QE architecture where MT and source are encoded together. This model is trained on DAs ranging 2017 to 2019 and then fine-tuned on DAs from MLQE-PE (the official DA from the QE shared task). The second model is the same multitask model used in the COMET-22 submission but without access to a reference translation. This means that this model is a multitask model trained on regression and sequence tagging. Both models are ensembled together using a hyperparameter search that weights different features extracted from these two QE models and combines them into a single score.

Huawei submitted several metrics to the shared task (Liu et al., 2022). **Cross-QE** is a submission based on the COMET-QE architecture. **HWTSC-Teacher-Sim** is a reference-free metric constructed by fine-tuning the multilingual Sentence BERT model: paraphrase-multilingual-mpnet-base-v2 (Reimers and Gurevych, 2019). **HWTSC-TLM** is a reference-free metric which only uses a target-side language model and only uses the system translations as input. **KG-BERTScore** is a reference-free machine translation evaluation metric, which incorporates a multilingual knowledge graph into BERTScore by linearly combining the results of BERTScore and bilingual named entity matching.

MATESE metrics (Perrella et al., 2022) leverage Transformer-based multilingual encoders to identify error spans in translations, and classify their severity between MINOR and MAJOR. The quality score returned for a translation is computed following the MQM error weighting introduced in Freitag et al. (2021a). MATESE is reference-based, while **MATESE-QE** is its reference-free version, with the source sentence used in place of the reference.

MEE (Mukherjee et al., 2020) is an automatic evaluation metric that leverages the similarity between embeddings of words in candidate and reference sentences to assess translation quality, focusing mainly on adequacy. Unigrams are matched based on their surface forms, root forms and meanings which aims to capture lexical, morphological and semantic equivalence. Semantic evaluation is achieved by using pretrained fasttext embeddings provided by Facebook to calculate the word similarity score between the candidate and reference words. MEE computes an evaluation score using three modules namely exact match, root match and synonym match. In each module, fmean-score is calculated using the harmonic mean of precision and recall by assigning more weightage to recall. The final translation score is obtained by taking average of fmean-scores from individual modules.

MEE2 and **MEE4** (Mukherjee and Shrivastava, 2022b) are improved versions of MEE, focusing on computing contextual and syntactic equivalences along with lexical, morphological and semantic similarity. The intent is to capture fluency and context of the MT outputs along with their adequacy. Fluency is captured using syntactic similarity and context is captured using sentence similarity leveraging sentence embeddings. The final sentence translation score is the weighted combination of three similarity scores: a) Syntactic Similarity achieved by modified BLEU score; b) Lexical, Morphological and Semantic Similarity: measured by explicit unigram matching similar to MEE score; c) Contextual Similarity: Sentence similarity scores are calculated by leveraging sentence embeddings of Language-Agnostic BERT models.

REUSE (Mukherjee and Shrivastava, 2022a) is a REference-free UnSupervised quality Estimation Metric. This is a bilingual untrained metric. It estimates the translation quality at chunk-level and sentence-level. Source and target sentence chunks are retrieved by using a multi-lingual chunker. Chunk-level similarity is computed by leveraging BERT contextual word embeddings and sentence similarity scores are calculated by leveraging sentence embeddings of Language-Agnostic BERT models. The final quality estimation score is obtained by mean pooling the chunk-level and sentence-level similarity scores.

6.3 Evaluation of Metrics

For all phenomena in ACES where we generated more than 1,000 examples, we randomly subsample 1,000 examples according to the per language pair distribution to include in the final challenge set to keep the evaluation of new metrics tractable.

We follow the evaluation of the challenge sets from the 2021 edition of the WMT metrics shared task (Freitag et al., 2021b) and report performance with Kendall's tau-like correlation. This metric measures the number of times a metric scores the good translation above the incorrect translation (concordant) and equal to or lower than the incorrect translation (discordant):

$$\tau = \frac{concordant - discordant}{concordant + discordant}$$

Ties are considered as discordant. Note that a higher τ indicates a better performance and that the values can range between -1 and 1.

7 Results

7.1 Phenomena-level Results

We start by providing a broad overview of metric performance on the different categories of phenomena. We compute Kendall's tau-like correlation scores (Section 6) for the 24 metrics which a) provide segment-level scores and b) provide scores for all language pairs and directions in ACES. We first compute the correlation scores for all of the individual phenomena and then take the average score over all phenomena in each of the nine top-level accuracy categories in ACES plus the fluency category punctuation (see Table 1).

The performance of the metrics varies greatly and there is no clear *winner* in terms of performance across all of the categories. There is also a high degree of variation in terms of metric performance when each category is considered in isolation. Whilst each of the categories proves challenging for at least one metric, some categories are more challenging than others. For example, looking at the average scores in the last row of Table 1, and without taking outliers into account, we might conclude that addition, undertranslation, real-world knowledge, and wrong language (all with average Kendall tau-like correlation of < 0.3) present more of a challenge than the other categories. On the other hand, for omission and do not translate (with an average Kendall tau-like correlation of > 0.7) metric performance is generally rather high.

We also observe variation in terms of the performance of metrics belonging to the baseline, reference-based, and reference-free groups. For example, the baseline metrics appear to struggle more on the overtranslation and undertranslation categories than the metrics belonging to the other groups. Reference-based metrics also appear to perform better overall on the untranslated category than the reference-free metrics. This makes sense as a comparison with the reference is likely to highlight tokens that ought to have been translated.

7.2 ACES Score

To analyse general, high-level, performance trends of the metrics on the ACES challenge set, we define a weighted combination of the top-level categories to derive a single score. We call this score the "ACES - Score":

$$\text{ACES} = sum\left\{\begin{array}{r} 5 * \tau_{\text{addition}} \\ 5 * \tau_{\text{omission}} \\ 5 * \tau_{\text{mistranslation}} \\ 1 * \tau_{\text{untranslated}} \\ 1 * \tau_{\text{do not translate}} \\ 5 * \tau_{\text{overtranslation}} \\ 5 * \tau_{\text{undertranslation}} \\ 1 * \tau_{\text{real-world knowledge}} \\ 1 * \tau_{\text{wrong language}} \\ 0.1 * \tau_{\text{punctuation}} \end{array}\right\} \quad (1)$$

The weights correspond to the values under the MQM framework that Freitag et al. (2021a) rec-

	addition	**omission**	**mistranslation**	**untranslated**	**do not translate**	**overtranslation**	**undertranslation**	**real-world knowledge**	**wrong language**	**punctuation**	**ACES-Score**
Examples	*999*	*999*	*24457*	*1300*	*100*	*1000*	*1000*	*2948*	*2000*	*1673*	
BLEU	0.748	0.435	-0.297	0.353	0.600	-0.838	-0.856	-0.768	0.661	0.638	-3.13
f101spBLEU	0.662	0.590	-0.132	0.660	0.940	-0.738	-0.826	-0.405	0.638	0.639	-0.33
f200spBLEU	0.664	0.590	-0.130	0.687	0.920	-0.752	-0.794	-0.394	0.658	0.648	-0.18
chrF	0.642	0.784	0.134	**0.781**	**0.960**	-0.696	-0.592	-0.294	**0.691**	0.743	3.57
BERTScore	**0.880**	0.750	0.283	0.767	**0.960**	-0.110	-0.190	0.031	0.563	**0.849**	10.47
BLEURT-20	0.437	0.810	0.396	0.748	0.860	0.200	0.014	0.401	0.533	0.649	11.90
COMET-20	0.437	0.808	0.336	0.748	0.900	0.314	0.112	0.267	0.033	0.706	12.06
COMET-QE	-0.538	0.397	0.417	0.135	0.120	0.622	0.442	0.322	-0.505	0.251	6.80
YiSi-1	0.770	0.866	0.325	0.730	0.920	-0.062	-0.076	0.110	0.431	0.734	11.38
COMET-22	0.333	0.806	0.546	0.536	0.900	0.690	0.538	0.574	-0.318	0.539	16.31
metricx_xl_DA_2019	0.395	0.852	0.521	0.722	0.940	0.692	0.376	**0.740**	0.521	0.670	17.17
metricx_xl_MQM_2020	-0.281	0.670	0.518	0.579	0.740	0.718	**0.602**	0.705	-0.126	0.445	13.08
metricx_xxl_DA_2019	0.303	0.832	0.558	0.762	0.920	0.572	0.246	0.691	0.250	0.630	15.24
metricx_xxl_MQM_2020	-0.099	0.534	0.579	0.651	0.880	**0.752**	0.552	0.712	-0.321	0.369	13.55
MS-COMET-22	-0.219	0.686	0.368	0.504	0.700	0.548	0.290	0.230	0.041	0.508	9.89
UniTE	0.439	0.876	0.467	0.571	0.920	0.496	0.302	0.624	-0.337	0.793	14.76
UniTE-ref	0.359	0.868	0.506	0.412	0.840	0.640	0.398	0.585	-0.387	0.709	15.38
COMETKiwi	0.361	0.830	**0.601**	0.230	0.780	0.738	0.574	0.582	-0.359	0.490	16.80
Cross-QE	0.163	0.876	0.505	-0.246	0.320	0.726	0.506	0.446	-0.374	0.455	14.07
HWTSC-Teacher-Sim	-0.031	0.495	0.381	-0.269	0.700	0.552	0.456	0.261	-0.021	0.271	9.97
HWTSC-TLM	-0.363	0.345	0.420	0.154	-0.040	0.544	0.474	0.071	-0.168	0.634	7.18
KG-BERTScore	0.790	0.812	0.447	-0.456	0.760	0.654	0.528	0.487	0.306	0.255	**17.28**
MS-COMET-QE-22	-0.177	0.678	0.401	0.388	0.240	0.518	0.386	0.248	-0.197	0.523	9.76
UniTE-src	0.285	**0.930**	0.565	-0.462	0.860	0.698	0.540	0.537	-0.417	0.733	15.68
Average	0.290	0.713	0.363	0.404	0.735	0.312	0.167	0.282	0.075	0.578	10.78

Table 1: Average Kendall's tau-like correlation results for the nine top level categories in the ACES ontology, plus the additional fluency category: punctuation. The horizontal lines delimit baseline metrics (top), participating reference-based metrics (middle) and participating reference-free metrics (bottom). The best result for each category is denoted by bold text with a green highlight. Note that *Average* is an average over averages. The last column shows the ACES-Score, a weighted sum of the correlations. The ACES-Score ranges from -29.1 (all phenomena have a correlation of -1) to 29.1 (all phenomena have a correlation of +1).

ommend for major (weight=5), minor (weight=1) and fluency/punctuation errors (weight=0.1). We determined that untranslated, do not translate and wrong language errors should be counted as minor errors because they can be identified automatically with language detection tools and should also be easy to spot in post-editing. We also include real-world knowledge under minor errors since we do not expect that current MT evaluation metrics have any notion of real-world knowledge and we do not want to punish them too severely if they do not perform well on this challenge set.

We caution that our weighting for the ACES-Score is not ideal, as some phenomena within a broad category might be more difficult than others. Still, we believe that an ACES-Score will be helpful to quickly identify changes in performance of a metric (e.g. following modifications), prior to conducting in-depth analyses at the category and sub-category levels. The ACES-Score ranges from -29.1 (all phenomena have a correlation of -1) to 29.1 (all phenomena have a correlation of +1).

The ACES-Score results can be seen in the last column of Table 1. Using the ACES-Score, we can see at a glance that the majority of the metrics submitted to the WMT 2022 shared task outperform the baseline metrics. Interestingly, many reference-free metrics also perform on par with reference-based metrics. The best performing metric is a reference-free metric, namely KG-BERTSCORE, closely followed by the reference-based metric METRICX_XL_DA_2019. Perhaps unsurprisingly, the worst performing metric is BLEU. However, we caution against making strong claims about which metrics perform *best* or *worst* on the challenge set based on this score alone. Instead, we recommend that ACES be used to highlight general trends as to what the outstanding issues are for MT evaluation metrics. More fine-grained analyses are reported in the following sections.

More generally, work on analysing system performance on ACES prompts the question: What is the definition of a good metric? One might consider that a *good* metric exhibits a strong correlation with human judgements on whether a translation is good/bad *and* assigns sufficiently different scores to a good vs. an incorrect translation. The latter criterion would provide evidence of the ability of the metric to discriminate reliably between good and incorrect translations, but it may be difficult to establish what this difference should be, especially without knowing to what degree the translations are good/bad without human judgements and because the scales of different metrics are not comparable. We leave an analysis of metrics' confidence on different error types for future work.

	disco.	**halluci.**	**other**
Examples	*3698*	*10270*	*10489*
BLEU	-0.048	-0.420	-0.251
f101spBLEU	0.105	-0.206	-0.153
f200spBLEU	0.094	-0.191	-0.149
chrF	0.405	-0.137	0.161
BERTScore	0.567	-0.058	0.362
BLEURT-20	0.695	0.142	0.402
COMET-20	0.641	0.016	0.399
COMET-QE	0.666	0.303	0.208
YiSi-1	0.609	0.019	0.368
COMET-22	0.682	0.461	0.542
metricx_xl_DA_2019	0.701	0.493	0.458
metricx_xl_MQM_2020	0.573	0.677	0.394
metricx_xxl_DA_2019	0.768	0.541	0.463
metricx_xxl_MQM_2020	0.716	**0.713**	0.392
MS-COMET-22	0.645	0.148	0.360
UniTE	0.746	0.322	0.424
UniTE-ref	**0.776**	0.396	0.437
COMETKiwi	0.733	0.493	**0.637**
Cross-QE	0.639	0.395	0.563
HWTSC-Teacher-Sim	0.594	0.296	0.330
HWTSC-TLM	0.756	0.306	0.151
KG-BERTScore	0.593	0.387	0.472
MS-COMET-QE-22	0.626	0.243	0.416
UniTE-src	0.172	0.463	0.551
Average	0.586	0.242	0.331

Table 2: Average Kendall's tau-like correlation results for the sub-level categories in mistranslation: **discourse-level**, **hallucination**, and **other** errors. The horizontal lines delimit baseline metrics (top), participating reference-based metrics (middle) and participating reference-free metrics (bottom). The best result for each category is denoted by bold text with a green highlight. Note that *Average* is an average over averages.

7.3 Mistranslation Results

Next, we drill down to the fine-grained categories of the largest category: *mistranslation*. We present metric performance on its sub-level categories in Table 2. Again, we find that performance on the different sub-categories is variable, with no clear *winner* among the metrics. The results suggest that hallucination phenomena are generally more challenging than discourse-level phenomena. Performance on the hallucination sub-category is poor overall, although it appears to be particularly challenging for the baseline metrics. We present additional, more fine-grained, performance analyses for individual phenomena in Section 8.

7.4 Language-level Results

	trained	**en-x**	**x-en**	**x-y**
Examples	8871	12695	17966	5815
BLEU	0.009	0.225	-0.370	-0.121
f101spBLEU	0.148	0.170	-0.290	-0.022
f200spBLEU	0.140	0.442	-0.286	-0.004
chrF	0.325	0.392	-0.047	0.098
BERTScore	0.479	0.031	0.173	0.125
BLEURT-20	0.541	0.327	0.280	0.257
COMET-20	0.495	0.379	0.278	0.121
COMET-QE	0.356	0.166	0.144	0.168
YiSi-1	0.476	0.520	0.185	0.150
COMET-22	0.599	0.486	0.554	0.355
metricx_xl_DA_2019	0.622	0.458	0.456	**0.551**
metricx_xl_MQM_2020	0.608	0.567	0.452	0.509
metricx_xxl_DA_2019	0.631	0.431	0.462	0.528
metricx_xxl_MQM_2020	0.605	**0.572**	0.487	0.502
MS-COMET-22	0.415	0.312	0.323	0.117
UniTE	0.635	0.452	0.406	0.283
UniTE-ref	0.619	0.313	0.413	0.305
COMETKiwi	0.620	0.510	**0.694**	0.468
Cross-QE	0.598	0.401	0.552	0.291
HWTSC-Teacher-Sim	0.497	0.357	0.352	0.149
HWTSC-TLM	0.538	0.519	0.167	0.194
KG-BERTScore	0.485	0.428	0.507	0.347
MS-COMET-QE-22	0.483	0.488	0.411	0.257
UniTE-src	**0.658**	0.445	0.582	0.328
MATESE	-0.281	n/a	n/a	n/a
MEE	-0.078	n/a	n/a	n/a
MEE2	0.340	n/a	n/a	n/a
MEE4	0.391	n/a	n/a	n/a
REUSE	0.430	n/a	n/a	n/a
MATESE-QE	-0.313	n/a	n/a	n/a

Table 3: Average Kendall's tau-like correlation results grouped by language pairs: trained language pairs (en-de, en-ru, zh-en), from English (en-x), into English (x-en) and language pairs not involving English (x-y). The horizontal lines delimit baseline metrics (top), all language pairs participating reference-based metrics (second), all language pairs participating reference-free metrics (third) and trained language pairs only metrics (bottom). The best result for each category is denoted by bold text with a green highlight.

Another possible way to evaluate the metrics' performance is not to look at the phenomena but rather at the results on different language pairs. Since ACES covers 146 language pairs and for some of these language pairs we only have very few examples, we decide to split this analysis into four main categories:

- **trained:** language pairs for which this year's WMT metrics shared task provided training material (en-de, en-ru and zh-en). This category also allows us to analyse the metrics that only cover these specific language pairs and not the full set of language pairs in ACES.
- **en-x:** language pairs where the source language is English.
- **x-en:** language pairs where the target language is English.
- **x-y:** all remaining language pairs, where neither the source language nor the target language are English.

Table 3 shows the results for all metrics. It is important to note that the results for different language pair categories cannot be directly compared because the examples and covered phenomena categories are not necessarily the same. However, we can compare metrics on each of the language pair groups individually. First, it can again be observed that most submitted metrics outperform the baseline metrics (first group). This shows that the field is advancing and MT evaluation metrics have improved since last year (i.e. 2021).

Interestingly, the six metrics that only scored the trained language pairs (last group in the table) do not outperform the other metrics on the "trained" category. Note, however, that the MEE* metrics and REUSE are unsupervised metrics and that the MATESE metrics only used MQM training data. Therefore, we cannot comment on creating metrics that are specific to a language pair would result in better metrics. In any case, our findings in Section 8.3.1 suggest that generalisation to unseen language pairs is generally quite good for the multilingual metrics which might be a more desirable property than increased performance on specific language pairs.

8 Analysis

Aside from high-level evaluations of which metrics perform best, we are mostly interested in metric-spanning weaknesses that we can identify using ACES. This section shows an analysis of three general questions that we aim to answer using ACES.

8.1 How sensitive are metrics to the source?

We designed our challenge sets for the type of ambiguous translation in a way that the correct translation candidate given an ambiguous reference can only be identified through the source sentence. Here, we present a targeted evaluation intended to provide some insights into how important the source is for different metrics. We exclude all metrics that do not take the source as input, all metrics

	since		**female**		**male**		**wsd**		
	causal	**temp.**	**anti.**	**pro.**	**anti.**	**pro.**	**freq.**	**infreq.**	**AVG**
Examples	*106*	*106*	*1000*	*806*	*806*	*1000*	*471*	*471*	*4766*
BERTScore	-0.434	0.434	-0.614	-0.216	0.208	0.618	0.214	-0.223	-0.001
COMET-20	-0.019	0.302	-0.622	-0.370	**0.586**	0.772	0.202	-0.079	0.097
COMET-22	-0.415	0.792	**0.940**	**1.000**	-0.628	0.374	**0.558**	**0.040**	**0.333**
metricx_xxl_DA_2019	-0.849	0.811	-0.944	-0.228	0.233	**0.942**	0.032	-0.028	-0.004
metricx_xxl_MQM_2020	-1.000	**1.000**	-0.878	0.002	-0.007	0.884	0.083	-0.100	-0.002
MS-COMET-22	-0.604	0.623	0.296	0.640	-0.342	0.046	0.316	-0.155	0.102
UniTE	**0.038**	-0.075	-0.890	-0.213	0.377	0.934	0.270	-0.223	0.027
COMET-QE	-1.000	**0.981**	0.450	0.871	-0.854	-0.382	0.244	-0.210	0.013
COMET-Kiwi	-0.245	0.943	0.964	0.978	0.794	**0.938**	0.648	**0.363**	**0.673**
Cross-QE	0.208	0.830	**0.976**	**0.995**	-0.337	0.364	**0.762**	0.355	0.519
HWTSC-Teacher-Sim	-0.453	0.717	0.916	0.772	-0.283	-0.360	0.295	0.079	0.210
KG-BERTScore	**0.453**	0.830	0.638	0.300	**0.968**	0.682	0.295	0.079	0.531
MS-COMET-QE-22	-0.283	0.792	-0.194	0.320	0.246	0.694	0.465	0.002	0.255
UniTE-src	-0.321	0.906	**0.976**	0.980	0.171	0.736	0.622	0.346	0.552

Table 4: Results on the challenge sets where the good translation can only be identified through the source sentence. Upper block: reference-based metrics, lower block: reference-free metrics. Best results for each phenomenon and each group of models is marked in bold and green and the average over all can be seen in the last column.

that do not cover all language pairs, and the smaller versions of METRIC-X (metricx_xl_DA_2019 and metricx_xl_MQM_2020) from this analysis. This leaves us with seven reference-based metrics and seven reference-free metrics. Table 4 shows the detailed results of each metric on the considered phenomena.

The most important finding is that the reference-free metrics generally perform much better on these challenge sets than the reference-based metrics. This indicates that reference-based metrics rely too much on the reference. Interestingly, most of the metrics that seem to ignore the source do not randomly guess the correct translation (which is a valid alternative choice when the correct meaning is not identified via the source) but rather they strongly prefer one phenomenon over the other. For example, several metrics show a gender bias either towards female occupation names (female correlations are high, male low) or male occupation names (vice versa). Likewise, most metrics prefer translations with frequent senses for the word-sense disambiguation challenge sets, although the difference between frequent and infrequent is not as pronounced as for gender.

Only metrics that look at the source and exhibit fewer such preferences can perform well on average on this collection of challenge sets. COMET-22 performs best out of the reference-based metrics and COMET-KIWI performs best of all reference-free metrics. It is noteworthy that there is still a considerable gap between these two models, suggesting that reference-based models should pay more attention to the source when a reference is ambiguous to reach the performance of reference-free metrics.

	corr. gain
BERTScore	0.002
COMET-20	0.060
COMET-22	**0.190**
metricx_xxl_DA_2019	0.012
metricx_xxl_MQM_2020	-0.016
MS-COMET-22	0.050
UniTE	0.042
COMET-QE	0.018
COMET-Kiwi	**0.338**
Cross-QE	0.292
HWTSC-Teacher-Sim	0.154
KG-BERTScore	0.154
MS-COMET-QE-22	0.196
UniTE-src	0.216

Table 5: Results on the real-world knowledge commonsense challenge set with reference-based metrics in the upper block and reference-free metrics in the lower block. The numbers are computed as the difference between the correlation with the subordinate clause in the source and the correlation without the subordinate clause in the source. Largest gains are bolded.

This finding is also supported by our real-world knowledge commonsense challenge set. If we compare the scores on the examples where the subor-

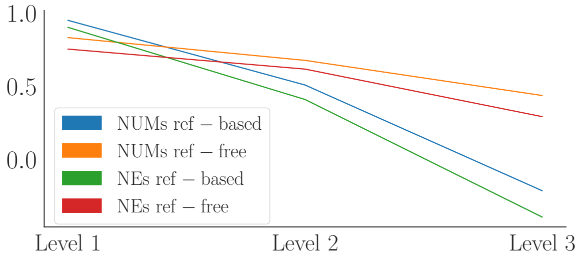


Figure 2: Decrease in correlation for reference-based and reference-free metrics on the named entity and number hallucination challenge sets.

dinate clauses are missing from both the source and the reference to the ones where they are only missing from the reference, we can directly see the effect of disambiguation through the source. The corresponding correlation gains are shown in Table 5. All reference-based model correlation scores improve less than most reference-free correlations when access to the subordinate clause is given through the source. This highlights again that reference-based metrics do not give enough weight to the source sentence.

8.2 How much do metrics rely on surface-overlap with the reference?

Another question we are interested in is whether neural reference-based metrics still rely on surface-level overlap with the reference. For this analysis, we use the dataset we created for hallucinated named entities and numbers. We take the average correlation for all reference-based metrics[16] and the average correlation of all reference-free metrics that cover all languages and plot the decrease in correlation with increasing surface-level similarity of the incorrect translation to the reference. The result can be seen in Figure 2.

We can see that on average reference-based metrics have a much steeper decrease in correlation than the reference-free metrics as the two translation candidates become more and more lexically diverse and the surface overlap between the incorrect translation and the reference increases. This indicates a possible weakness of reference-based metrics: If one translation is lexically similar to the reference but contains a grave error while others are correct but share less surface-level overlap with the reference, the incorrect translation may still be preferred.

	reference-based	reference-free
hallucination	-0.22 ± 0.16	+0.04 ± 0.07
overly-literal	-0.32 ± 0.16	+0.12 ± 0.09
untranslated	-0.44 ± 0.18	+0.03 ± 0.24

Table 6: Average correlation difference and standard deviation between the challenge sets with reference-copied good translations and the challenge sets with the synonymous good translations.

We also show that this is the case for the challenge set where we use an adversarial paraphrase from PAWS-X that shares a high degree of lexical overlap with the reference but does not have the same meaning as an incorrect translation. On average, the reference-based metrics only reach a correlation of 0.05 ± 0.12 on this challenge set, whereas the reference-free metrics reach a correlation of 0.23 ± 0.15. This shows that reference-based metrics are less robust when the incorrect translation has high lexical overlap with the reference.

Finally, we can also see a clear effect of surface-level overlap with the source on three real error challenge sets where we have different versions of the good translation: some where the error was corrected with the corresponding correct token from the reference and some where the error was corrected with a synonym for the correct token from the reference. As seen in Table 6, the reference-based metrics show a much larger difference in correlation between the challenge sets with reference-copied good translations and the challenge sets with the synonymous good translations, than the reference-free metrics. For example, for the hallucination test set, reference-free metrics have very similar average performance when the good translation contains the same word as the reference vs. when it contains a synonym (δ of +0.04). On the other hand, the reference-based metrics lose on average -0.22 in correlation when the good translation contains the synonym rather than the same word as the reference. Based on all of these results, we conclude that even though state-of-the-art reference-based MT evaluation metrics are not only reliant on surface-level overlap anymore, such overlap still considerably influences their predictions.

8.3 Do multilingual embeddings help design better metrics?

As the community moves towards building metrics that use multilingual encoders, we investigate if some (un)desirable properties of multilingual em-

[16] Excluding surface-level baseline metrics: BLEU, SPBLEU and CHRF.

beddings are propagated in these metrics.

8.3.1 Zero-shot Performance

Similar to Kocmi et al. (2021), we investigate whether there is a difference in the performance of metrics on our challenge sets when evaluated on non-WMT language pairs *i.e.* language pairs unseen during the training of the metrics. For this analysis, we include only those metrics for which the training data consisted of some combination of WMT human evaluation data. As different metrics used data from different years, we consider an intersection of languages across these years as WMT language pairs. For a fair comparison, we consider a subset of examples from those phenomena where we have least 100 examples in WMT languages and 100 examples in non-WMT languages, irrespective of the number of examples per individual language pair. We report some of the phenomena in Table 7, where metrics are compared in terms of the correlation difference between the performance on WMT and non-WMT language pairs (see Appendix A.3 for the original WMT and non-WMT correlation scores and the list of language pairs).

	antonym-replacement	real-world knowledge commonsense	nonsense
Examples	*131*	*201*	*239*
BERTScore	0.032	-0.054	1.469
BLEURT-20	0.032	0.201	0.350
COMET-20	0.048	0.067	1.021
COMET-QE	-0.048	-0.188	-0.294
COMET-22	0.080	0.027	0.531
metricx_xl_DA_2019	-0.032	-0.054	0.434
metricx_xl_MQM_2020	-0.048	-0.094	0.182
metricx_xxl_DA_2019	0.016	-0.040	0.266
metricx_xxl_MQM_2020	0.064	-0.067	0.196
UniTE-ref	-0.032	0.013	0.238
UniTE	0.080	0.000	0.643
COMETKiwi	0.048	-0.027	0.042
Cross-QE	0.064	0.188	0.182
HWTSC-Teacher-Sim	0.208	0.081	0.350
UniTE-src	0.096	0.161	-0.028

Table 7: Correlation difference between the performance of WMT and non-WMT language pairs reported for trained metrics across a subset of examples. $\delta = \tau_{\text{WMT}} - \tau_{\text{non WMT}}$. WMT language pairs consist of a subset of languages seen during training of the metrics, while non-WMT language pairs are unseen. Results show that the metrics are able to generalise to unseen languages.

We draw similar conclusions to Kocmi et al. (2021), namely that trained metrics are not overfitted to the WMT language pairs. We observe that the median difference of τ between WMT and non-WMT language pairs is 0.056, indicating a good generalisation to unseen languages. We still note that performance on the phenomena is variable when we compare the results on WMT language pairs versus non-WMT language pairs. In the case of real-world knowledge commonsense, performance is slightly better on the non-WMT language pairs[17], while the opposite is (generally) true for the antonym replacement and, especially, the nonsense phenomena for certain metrics. Further analysis is required to better understand metric behaviour on zero-shot language pairs, especially considering that some of the analysed non-WMT language pairs have a target language that is also the target language in at least one of the WMT language pairs (e.g. English).

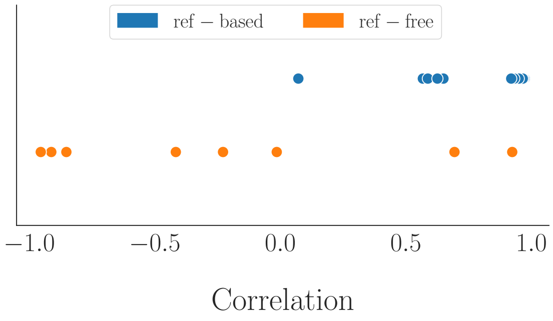


Figure 3: Correlation of reference-based metrics (blue) and reference-free metrics (orange) on the sentence-level untranslated test challenge set.

8.3.2 Language Dependent Representations

Multilingual models often learn cross-lingual representations by abstracting away from language-specific information (Wu and Dredze, 2019). We are interested in whether the representations are still language-dependent in neural MT evaluation metrics which are trained on such models. For this analysis, we look at the sentence-level untranslated text challenge set (see Figure 3) and wrong language phenomena (see Table 1). We only consider metrics that provided scores for examples in all language pairs.

Figure 3 shows the correlations for all reference-based and reference-free metrics. Unsurprisingly, some reference-free metrics struggle considerably on this challenge set and almost always prefer the copied source to the real translation. The representations of the source and the incorrect translation are identical, leading to a higher surface and embedding similarity, and thus a higher score. We do, however, find some exceptions to this trend

[17] We also observe better performance on non-WMT language pairs for the similar language high phenomenon.

- COMET-Kiwi and MS-COMET-QE-22 both have a high correlation on sentence-level untranslated text. This suggests that these metrics could have learnt language-dependent representations.

Most reference-based metrics have good to almost perfect correlation and can identify the copied source quite easily. As reference-based metrics tend to ignore the source (see Section 8.2), the scores are based on the similarity between the reference and the MT output. In this challenge set, the similarity between the good-translation and the reference is likely to be higher than the incorrect-translation and the reference. The former MT output is in the same language as the reference and will have more surface level overlap. We believe the reference here acts as grounding.

However, this grounding property of the reference is only robust when the source and reference languages are dissimilar, as is the case with language pairs in the sentence-level untranslated text challenge set. We find that reference-based metrics struggle on wrong language phenomena (see Table 1) where the setup is similar, but now the incorrect translation and the reference are from similar languages (e.g. one is in Hindi and the other is in Marathi). Naturally, there will be surface level overlap between the reference and both the good-translation and the incorrect-translation. For example, both Marathi and Hindi use named entities with identical surface form, and so these will appear in the reference and also in both the good-translation and the incorrect-translation. Thus, the semantic content drives the similarity scores between the MT outputs and the references. It is possible that the human translation in the similar language (labelled as the incorrect-translation) has a closer representation to the human reference because in the MT output (labelled as the good-translation) some semantic information may be lost. We leave further investigation of this for future work.

While multilingual embeddings help in effective zero-shot transfer to new languages, some properties of the multilingual representation space may need to be altered to suit the task of machine translation evaluation.

9 Recommendations

Based on the metrics results on ACES and our analysis, we derived the following list of recommendations for future MT evaluation metric development:

No metric to rule them all: Both the evaluation on phenomena and on language pair categories in Section 7 showed that there is no single best-performing metric. This divergence is likely to become even larger if we evaluate metrics on different domains. For future work on MT evaluation, it may be worthwhile thinking about how different metrics can be combined to make robust decisions as to which is the best translation. This year's submissions to the metrics shared task already suggest that work in that direction is ongoing as some groups submitted metrics that combined ensembles of models or multiple components (COMET-22, COMET-Kiwi, KG-BERTScore, MEE*, REUSE).

The source matters: Our analysis in Section 8.1 highlighted that many reference-based metrics that take the source as input do not consider it enough. Cases where the correct translation can only be identified through the source are currently better handled by reference-free metrics. This is a serious shortcoming of reference-based metrics and should be addressed in future research, also considering that many reference-based metrics do not even take the source as input.

Surface-overlap still prevails: In Section 8.2, we showed that despite moving beyond only surface-level comparison to the reference, most reference-based metric scores are still considerably influenced by surface-level overlap. We expect future metrics to use more lexically diverse references in their training regime to mitigate this issue.

Multilingual embeddings are not perfect: Some properties of multilingual representations, especially, being language-agnostic, can result in undesirable effects on MT evaluation (Section 8.3). It could be helpful for future metrics to incorporate strategies to explicitly model additional language-specific information.

10 Conclusion

We presented ACES, a translation accuracy challenge set based on the MQM ontology. ACES consists of 36,476 examples covering 146 language pairs and representing challenges from 68 phenomena. We used ACES to evaluate the baseline and submitted metrics from the WMT 2022 metrics shared task. Our overview of metric performance at the phenomena and language levels in Section 7 reveals that there is no single best-performing metric. The more fine-grained analyses in Section 8 highlight that 1) many reference-based metrics that take the source as input do not consider it enough, 2) most reference-based metric scores are still considerably influenced by surface overlap with the reference, and 3) the use of multilingual embeddings can have undesirable effects on MT evaluation.

We recommend that these shortcomings of existing metrics be addressed in future research, and that metric developers should consider a) combining metrics with different strengths, e.g. in the form of ensemble models, b) developing metrics that give more weight to the source and less to surface-level overlap with the reference, and c) incorporating strategies to explicitly model additional language-specific information (rather than simply relying on multilingual embeddings).

We have made ACES publicly available and hope that it will provide a useful benchmark for MT evaluation metric developers in the future.

Limitations

The ACES challenge set exhibits a number of biases. Firstly, there is greater coverage in terms of phenomena and number of examples for the en-de and en-fr language pairs. This is in part due to the manual effort required to construct examples for some phenomena, in particular those belonging to the discourse-level and real-world knowledge categories. Further, our choice of language pairs is also limited to the ones available in XLM-R. Secondly, ACES contains more examples for those phenomena for which examples could be generated automatically, compared to those that required manual construction/filtering. Thirdly, some of the automatically generated examples require external libraries which are only available for a few languages (e.g. Multilingual Wordnet). Fourthly, the focus of the challenge set is on accuracy errors. We leave the development of challenge sets for fluency errors to future work.

As a result of using existing datasets as the basis for many of the examples, errors present in these datasets may be propagated through into ACES. Whilst we acknowledge that this is undesirable, in our methods for constructing the *incorrect translation* we aim to ensure that the quality of the *incorrect translation* is always worse than the corresponding *good translation*.

The results and analyses presented in the paper exclude those metrics submitted to the WMT 2022 metrics shared task that provide only system-level outputs. We focus on metrics that provide segment-level outputs as this enables us to provide a broad overview of metric performance on different phenomenon categories and to conduct fine-grained analyses of performance on individual phenomena. For some of the fine-grained analyses, we apply additional constraints based on the language pairs covered by the metrics, or whether the metrics take the source as input, to address specific questions of interest. As a result of applying some of these additional constraints, our investigations tend to focus more on high and medium-resource languages than on low-resource languages. We hope to address this shortcoming in future work.

Ethics Statement

Some examples within the challenge set exhibit biases, however this is necessary in order to expose the limitations of existing metrics. Wherever external help was required in verifying translations, the annotators were compensated at a rate of £15/hour. Our challenge set is based on publicly available datasets and will be released for future use.

Acknowledgements

We thank the organisers of the WMT 2022 Metrics task for setting up this shared task and for their feedback throughout the process, and the shared task participants for scoring our challenge sets with their systems. We are grateful to Stephanie Droop, Octave Mariotti, and Kenya Murakami for helping us with the annotations. We thank the StatMT group at Edinburgh, especially Alexandra Birch, Barry Haddow, and Ulrich Germann, and the attendees at the MT Marathon 2022 for their valuable feedback. We thank Janis Goldzycher, Mark Steedman, Rico Sennrich, and the anonymous reviewers for their insightful comments and suggestions. This work was supported in part by the UKRI Centre for Doctoral Training in Natural Language Processing, funded by the UKRI (grant EP/S022481/1) and the University of Edinburgh (Moghe), by the Swiss National Science Foundation (project MUTAMUR; no. 176727) (Amrhein) and by the ERC H2020 Advanced Fellowship GA 742137 SEMANTAX (Guillou). We also thank Huawei for their support (Moghe).

References

Chantal Amrhein and Rico Sennrich. 2022. Identifying weaknesses in machine translation metrics through minimum bayes risk decoding: A case study for COMET. In *2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, Online. Association for Computational Linguistics.

Eleftherios Avramidis, Vivien Macketanz, Arle Lommel, and Hans Uszkoreit. 2018. Fine-grained evaluation of quality estimation for machine translation based on a linguistically motivated test suite. In *Proceedings of the AMTA 2018 Workshop on Translation Quality Estimation and Automatic Post-Editing*, pages 243–248, Boston, MA. Association for Machine Translation in the Americas.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus phrase-based machine translation quality: a case study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 257–267, Austin, Texas. Association for Computational Linguistics.

Sheila Castilho, Joss Moorkens, Federico Gaspari, Iacer Calixto, John Tinsley, and Andy Way. 2017. Is neural machine translation the new state of the art? *The Prague Bulletin of Mathematical Linguistics*, 108:109–120.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Verna Dankers, Christopher Lucas, and Ivan Titov. 2022. Can transformer be too compositional? analysing idiom processing in neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3608–3626, Dublin, Ireland. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Denis Emelin and Rico Sennrich. 2021. Wino-X: Multilingual Winograd schemas for commonsense reasoning and coreference resolution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8517–8532, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021a. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021b. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Liane Guillou and Christian Hardmeier. 2016. PROTEST: A test suite for evaluating pronouns in machine translation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 636–643, Portorož, Slovenia. European Language Resources Association (ELRA).

Liane Guillou, Christian Hardmeier, Ekaterina Lapshinova-Koltunski, and Sharid Loáiciga. 2018. A pronoun test suite evaluation of the English–German MT systems at WMT 2018. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 570–577, Belgium, Brussels. Association for Computational Linguistics.

Michael Hanna and Ondřej Bojar. 2021. A fine-grained analysis of BERTScore. In *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517, Online. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multitask benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Pierre Isabelle, Colin Cherry, and George Foster. 2017. A challenge set approach to evaluating machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2496, Copenhagen, Denmark. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Margaret King and Kirsten Falkedal. 1990. Using test suites in evaluation of machine translation systems. In *COLING 1990 Volume 2: Papers presented to the 13th International Conference on Computational Linguistics*.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.

Tom Kocmi, Hitokazu Matsushita, and Christian Federmann. 2022. MS-COMET: More and Better Human Judgements Improve Metric Performance. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Majid Laali and Leila Kosseim. 2017. Improving discourse relation projection to build discourse annotated corpora. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 407–416, Varna, Bulgaria. INCOMA Ltd.

Ekaterina Lapshinova-Koltunski, Christian Hardmeier, and Pauline Krielke. 2018. ParCorFull: a parallel corpus annotated with full coreference. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Yitong Li, Trevor Cohn, and Timothy Baldwin. 2017. BIBI system description: Building with CNNs and breaking with deep reinforcement learning. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 27–32, Copenhagen, Denmark. Association for Computational Linguistics.

Yilun Liu, Xiaosong Qiao, Zhanglin Wu, Su Chang, Min Zhang, Yanqing Zhao, shimin tao Song Peng, Hao Yang, Ying Qin, Jiaxin Guo, Minghan Wang, Yinglu Li, Peng Li, and Xiaofeng Zhao. 2022. Partial Could Be Better Than Whole: HW-TSC 2022 Submission for the Metrics Shared Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Arle Lommel, Aljoscha Burchardt, and Hans Uszkoreit. 2014. Multidimensional quality metrics (mqm): A framework for declaring and describing translation quality metrics. *Tradumàtica: tecnologies de la traducció*, 0:455–463.

Taylor Mahler, Willy Cheung, Micha Elsner, David King, Marie-Catherine de Marneffe, Cory Shain, Symon Stevens-Guille, and Michael White. 2017. Breaking NLP: Using morphosyntax, semantics, pragmatics and world knowledge to fool sentiment analysis systems. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 33–39, Copenhagen, Denmark. Association for Computational Linguistics.

Richard T McCoy and Tal Linzen. 2019. Non-entailed subsequences as a challenge for natural language inference. *Proceedings of the Society for Computation in Linguistics (SCiL)*, pages 358–360.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Ananya Mukherjee, Hema Ala, Manish Shrivastava, and Dipti Misra Sharma. 2020. Mee : An automatic metric for evaluation using embeddings for machine translation. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 292–299.

Ananya Mukherjee and Manish Shrivastava. 2022a. REUSE: REference-free UnSupervised quality Estimation Metric. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Ananya Mukherjee and Manish Shrivastava. 2022b. Unsupervised Embedding-based Metric for MT Evaluation with Improved Human Correlation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Stefano Perrella, Lorenzo Proietti, Alessandro Scirè, Niccolò Campolungo, and Roberto Navigli. 2022. Machine Translation Evaluation as a Sequence Tagging Problem. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Maja Popović and Sheila Castilho. 2019. Challenge test sets for MT evaluation. In *Proceedings of Machine Translation Summit XVII: Tutorial Abstracts*, Dublin, Ireland. European Association for Machine Translation.

Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. 2019. The MuCoW test suite at WMT 2019: Automatically harvested multilingual contrastive word sense disambiguation test sets for machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 470–480, Florence, Italy. Association for Computational Linguistics.

Abhilasha Ravichander, Siddharth Dalmia, Maria Ryskina, Florian Metze, Eduard Hovy, and Alan W Black. 2021. NoiseQA: Challenge set evaluation for user-centric question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2976–2992, Online. Association for Computational Linguistics.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 Submission for the Metrics Shared Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Laura Rimell, Stephen Clark, and Mark Steedman. 2009. Unbounded dependency recovery for parser evaluation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 813–821, Singapore. Association for Computational Linguistics.

Guido Rocchietti, Flavia Achena, Giuseppe Marziano, Sara Salaris, and Alessandro Lenci. 2021. Fancy: A diagnostic data-set for nli models. In *Proceedings of the Eighth Italian Conference on Computational Linguistics (CLiC-it)*.

Rachel Rudinger, Chandler May, and Benjamin Van Durme. 2017. Social bias in elicited natural language inferences. In *Proceedings of the First ACL*

Workshop on Ethics in Natural Language Processing, pages 74–79, Valencia, Spain. Association for Computational Linguistics.

Thibault Sellam, Amy Pu, Hyung Won Chung, Sebastian Gehrmann, Qijun Tan, Markus Freitag, Dipanjan Das, and Ankur Parikh. 2020. Learning to evaluate translation beyond English: BLEURT submissions to the WMT metrics 2020 shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 921–927, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2888–2913, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Noah A. Smith. 2012. Adversarial evaluation for models of natural language. *CoRR*, abs/1207.0245.

Lucia Specia, Zhenhao Li, Juan Pino, Vishrav Chaudhary, Francisco Guzmán, Graham Neubig, Nadir Durrani, Yonatan Belinkov, Philipp Koehn, Hassan Sajjad, Paul Michel, and Xian Li. 2020. Findings of the WMT 2020 shared task on machine translation robustness. In *Proceedings of the Fifth Conference on Machine Translation*, pages 76–91, Online. Association for Computational Linguistics.

Ieva Staliūnaitė and Ben Bonfil. 2017. Breaking sentiment analysis of movie reviews. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 61–64, Copenhagen, Denmark. Association for Computational Linguistics.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy. Association for Computational Linguistics.

Antonio Toral and Víctor M. Sánchez-Cartagena. 2017. A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1063–1073, Valencia, Spain. Association for Computational Linguistics.

Jannis Vamvas and Rico Sennrich. 2021. Contrastive conditioning for assessing disambiguation in MT: A case study of distilled bias. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10246–10265, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jannis Vamvas and Rico Sennrich. 2022. As little as possible, as much as necessary: Detecting over- and undertranslations with contrastive conditioning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 490–500, Dublin, Ireland. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Lucas Nunes Vieira, Minako O'Hagan, and Carol O'Sullivan. 2021. Understanding the societal impacts of machine translation: a critical review of the literature on medical and legal use cases. *Information, Communication & Society*, 24(11):1515–1532.

Yu Wan, Keqin Bao, Dayiheng Liu, Baosong Yang, Derek F. Wong, Lidia S. Chao, Wenqiang Lei, and Jun Xie. 2022. Alibaba-Translate China's Submission for WMT2022 Metrics Shared Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

Jianing Zhou, Hongyu Gong, and Suma Bhat. 2021. PIE: A parallel idiomatic expression corpus for idiomatic sentence generation and paraphrasing. In *Proceedings of the 17th Workshop on Multiword Expressions (MWE 2021)*, pages 33–48, Online. Association for Computational Linguistics.

A Appendix

A.1 Language Codes

Code	Language	Code	Language
af	Afrikaans	ja	Japanese
ar	Arabic	ko	Korean
be	Belarusian	lt	Lithuanian
bg	Bulgarian	lv	Latvian
ca	Catalan	mr	Marathi
cs	Czech	nl	Dutch
da	Danish	no	Norwegian
de	German	pl	Polish
el	Greek	pt	Portuguese
en	English	ro	Romanian
es	Spanish	ru	Russian
et	Estonian	sk	Slovak
fa	Persian	sl	Slovenian
fi	Finnish	sr	Serbian
fr	French	sv	Swedish
ga	Irish	sw	Swahili
gl	Galician	ta	Tamil
he	Hebrew	th	Thai
hi	Hindi	tr	Turkish
hr	Croatian	uk	Ukranian
hu	Hungarian	ur	Urdu
hy	Armenian	vi	Vietnamese
id	Indonesian	wo	Wolof
it	Italian	zh	Chinese

Table 8: ISO 2-Letter language codes of the languages included in the challenge set

A.2 Permitted Unit Conversions

We allow the following unit conversions for the challenge set that covers such errors:

Distance:

- miles → metres
- kilometres → miles
- kilometres → metres
- metres → feet
- metres → yards
- feet → metres
- feet → yards
- centimetres → inches
- centimetres → millimetres
- inches → centimetres

- inches → millimetres
- millimetres → centimetres
- millimetres → inches
- millimetres → inches

Speed:

- miles per hour → kilometres per hour
- kilometres per hour → miles per hour
- kilometres per second → miles per second
- miles per second → kilometres per second

Time:

- hours → minutes
- minutes → seconds
- seconds → minutes
- days → hours
- months → weeks
- weeks → days

Volume:

- barrels → gallons
- barrels → litres
- gallons → barrels
- gallons → litres

Weight:

- kilograms → grams
- kilograms → pounds
- grams → ounces
- ounces → grams

Area:

- square kilometres → square miles

A.3 Zero Shot Performance Scores

Table 9 contains the Kendall tau-like correlation scores for neural metrics on WMT language pairs (a subset of those seen during training) and non-WMT language pairs (unseen), for three phenomena: antonym replacement, real-world knowledge commonsense, and nonsense. The table contains the complete set of scores, and complements Table 7, which reports only the difference between the non-WMT and WMT correlation scores. See Section 8.3.1 on zero-shot performance. We shall now list the language pairs across the different phenomena:

Antonym Replacement

WMT:	de-en
non-WMT:	ko-en, es-en

Real-world Knowledge - Commonsense

WMT:	de-en, ru-en, en-ru, en-de
non-WMT:	ru-de, fr-ru, ru-fr, de-ru

Nonsense

WMT:	de-en
non-WMT:	fr-ja, ko-ja, en-ko, ko-en

Note that the subset of examples used in this analysis only consists of mid/high resource language pairs; investigation into the performance on low-resource languages is left for future work.

A.4 Distribution of Examples Across Language Pairs

Table 10 contains the total number of examples per language pair in the challenge set. As can be seen in the table, the distribution of examples is variable across language pairs. The dominant language pairs are: en-de, de-en, and fr-en.

A.5 Distribution of Language Pairs Across Phenomena

Table 11 contains the list of language pairs per phenomena in the challenge set. As can be seen in the table, the distribution of language pairs is variable across phenomena. Addition and omission have the highest variety of language pairs. en-de is the most frequent language pair across all phenomena.

	antonym-replacement		real-world knowledge -commonsense		nonsense	
	WMT	Non-WMT	WMT	Non-WMT	WMT	Non-WMT
BERTScore	-0.376	-0.408	0.007	0.060	0.790	-0.678
BLEURT-20	0.024	-0.008	0.396	0.195	-0.273	-0.622
COMET-20	0.152	0.104	0.087	0.020	0.706	-0.315
COMET-QE	0.616	0.664	0.168	0.356	0.245	0.538
COMET-22	0.744	0.664	0.584	0.557	0.706	0.175
metricx_xl_DA_2019	0.728	0.760	0.570	0.624	0.790	0.357
metricx_xl_MQM_2020	0.888	0.936	0.517	0.611	0.944	0.762
metricx_xxl_DA_2019	0.312	0.296	0.718	0.758	0.706	0.441
metricx_xxl_MQM_2020	0.696	0.632	0.691	0.758	0.930	0.734
UniTE-ref	0.664	0.696	0.409	0.396	0.091	-0.147
UniTE	0.632	0.552	0.409	0.409	0.441	-0.203
COMETKiwi	0.744	0.696	0.745	0.772	0.510	0.469
Cross-QE	0.680	0.616	0.638	0.450	0.720	0.538
HWTSC-Teacher-Sim	0.504	0.296	0.248	0.168	0.930	0.580
UniTE-src	0.776	0.680	0.651	0.490	0.524	0.552

Table 9: Zero-shot performance of neural metrics on three phenomena to measure the ability of metrics to generalise to new language pairs. WMT language pairs consist of a subset of languages seen during training of the metrics, while non-WMT language pairs are unseen. Results show that the metrics are able to generalise to unseen languages.

src ↓ / tgt →	af	ar	be	bg	ca	cs	da	de	el	en	es	et	fa	fi	fr	ga	gl	he	hi	hr	hu	hy	id	it	ja	ko	lt	lv	mr	nl	no	pl	pt	ro	ru	sk	sl	sr	sv	sw	ta	th	tr	uk	ur	vi	wo	zh
af										96			25																																			
ar										361					102				17																													
be										67																																						
bg										393																	40																					
ca										79	175																																					
cs										85																																						
da										83																																						
de										4163	84				394										113	63									104													75
el										387																																						
en	25	5	6	15	347	368	46	6964	21		725	25	20	12	800		16	18	343	27	44	3	31	10	430	545	17	19	52	50	53	349	44	46	698	27	45	15	39		1		10	16	10	25		333
es					88			64		1263					125										117	74																						67
et										70																																						
fa	16									85																																						
fi										79																																						
fr								683		2868	78														403	59			344						46													61
ga										17																																						
gl										70																																						
he										59																													51									
hi		8								367																																						
hr										81																		29																				
hu										53																																						
hy										48																																				13		
id										63																																						
it										801																																						
ja								60		912	67				122											163																						74
ko								70		1004	72				110										358																							73
lt				28						68																																						
lv										61										24																												
mr										63																																						
nl										73																																						
no										53																																						
pl										65																			111							58												
pt										89																												40										
ro										91																																						
ru								106		472	87				42																																	
sk										54																						17																
sl										69																																						
sr										64																							54															
sv										79								28																														
sw										327																																						
ta										39																																						
th										299																																						
tr										386																																						
uk										77																																						
ur										372																																						
vi										391												3																										
wo										11																																						
zh								150		1209	59				113										128	80																						

Table 10: Number of examples per language pair. Rows: source language; Columns: target language

phenomena	language pairs	phenomena	language pair
ambiguous-translation-wrong-discourse-connective-causal hallucination-unit-conversion-unit-matches-ref	fr-en, de-en	hallucination-real-data-vs-ref-word	en-de, de-en, fr-de
ambiguous-translation-wrong-discourse-connective-while-contrast	fr-en	hallucination-real-data-vs-synonym	en-mr, de-en, en-de, fr-de
ambiguous-translation-wrong-discourse-connective-while-temporal	fr-en	untranslated-vs-ref-word	en-de, de-en, fr-de
ambiguous-translation-wrong-gender-female-anti	fr-en, de-en, it-en	untranslated-vs-synonym	en-de, de-en, fr-de
ambiguous-translation-wrong-gender-male-anti	fr-en, de-en, it-en	modal_verb:deletion	de-en
ambiguous-translation-wrong-gender-male-pro	fr-en, de-en, it-en	modal_verb:substitution	de-en
ambiguous-translation-wrong-sense-frequent	en-de, en-ru	nonsense	ko-en, ko-ja, en-ko, fr-ja, de-en
ambiguous-translation-wrong-sense-infrequent	en-de, en-ru	ordering-mismatch	en-de, de-en, fr-de
anaphoric_group_it-they:deletion	en-de	overly-literal-vs-correct-idiom	en-de, de-en
anaphoric_group_it-they:substitution	en-de	overly-literal-vs-explanation	en-de, de-en
anaphoric_intra_non-subject_it:deletion	en-de	overly-literal-vs-ref-word	en-de, de-en, fr-de
anaphoric_intra_non-subject_it:substitution	en-de	overly-literal-vs-synonym	en-mr, de-en, en-de, fr-de
anaphoric_intra_subject_it:deletion	en-de	pleonastic_it:deletion	en-de
anaphoric_intra_subject_it:substitution	en-de	pleonastic_it:substitution_pro_trans_different_to_ref	en-de
anaphoric_intra_they:deletion	en-de	punctuation:deletion_all	en-de
anaphoric_intra_they:substitution	en-de	punctuation:deletion_commas	en-de
anaphoric_singular_they:deletion	en-de	punctuation:deletion_quotes	en-de
anaphoric_singular_they:substitution	en-de	punctuation:statement-to-question	en-de
antonym-replacement	fr-en, ko-en, ja-en, es-en, zh-en, de-en	real-world-knowledge-entailment	en-de, de-en
similar-language-high	en-hi, en-cs, en-es	real-world-knowledge-hypernym-vs-distractor	en-de, de-en
similar-language-low	fr-mr, en-pl, en-ca	real-world-knowledge-hypernym-vs-hyponym	en-de, de-en
coreference-based-on-commonsense	en-de, en-ru, en-fr	real-world-knowledge-synonym-vs-antonym	en-de, de-en
hallucination-named-entity-level-1 hallucination-named-entity-level-2 hallucination-named-entity-level-3 hallucination-number-level-1 hallucination-number-level-2 hallucination-number-level-3	en-de, ja-de, en-ko, de-zh, ja-en, es-de, fr-en, es-ko, ko-ja, es-ja, de-ja, zh-es, fr-zh, fr-ja, es-en, fr-ko, zh-en, ko-de, ko-es, de-ko, ko-en, fr-es, ja-es, ja-ko, zh-fr, en-es, de-en, ja-fr, ko-zh, en-fr, de-fr, ko-fr, es-fr, zh-ko, fr-de, ja-zh, de-es, es-zh, en-ja, zh-de, en-zh, zh-ja	hyponym-replacement hypernym-replacement	fr-en, ko-en, ja-en, es-en, zh-en, de-en
lexical-overlap	fr-en, en-fr, de-fr, ko-en, es-ja, ja-en, ko-fr, es-fr, ko-ja, de-ja, zh-en, ja-fr, zh-fr, en-ja, es-en, fr-ja, de-en, zh-ja	xnli-addition-contradiction xnli-addition-neutral xnli-omission-contradiction xnli-omission-neutral	fr-en, vi-en, sw-en, tr-en, zh-en, ru-en, bg-en, el-en, th-en, es-en, hi-en, de-en, ar-en, ur-en
hallucination-unit-conversion-amount-matches-ref hallucination-unit-conversion-unit-matches-ref	et-en, wo-en, da-en, no-en, uk-en, ta-en, fi-en, pl-en, ja-en, hy-en, ur-en, hr-en, fr-en, lt-en, tr-en, he-en, bg-en, ro-en, sv-en, ru-en, es-en, nl-en, zh-en, hu-en, be-en, lv-en, ko-en, ga-en, sk-en, af-en, sl-en, sr-en, ca-en, de-en, mr-en, id-en, vi-en, gl-en, pt-en, fa-en, hi-en, el-en, ar-en, it-en, cs-en	hallucination-date-time	en-de, et-en, ca-es, en-et, hr-lv, da-en, no-en, uk-en, fi-en, en-da, ta-en, pl-en, ja-en, en-hr, hy-en, ur-en, fr-en, hr-en, lt-en, sr-pt, en-sv, tr-en, en-no, en-sl, he-en, pl-sk, ru-en, ro-en, sv-en, en-lt, es-en, en-nl, nl-en, bg-en, he-sv, zh-en, hu-en, be-en, lv-hr, lv-en, bg-lt, en-ro, sk-pl, ko-en, ga-en, sk-en, af-en, sl-en, en-hu, sr-en, en-es, ca-en, en-sk, de-en, mr-en, id-en, vi-en, gl-en, en-fr, de-fr, pt-en, fr-de, en-pt, fa-en, hi-en, el-en, ar-en, it-en, en-pl, cs-en
commonsense-only-ref-ambiguous commonsense-src-and-ref-ambiguous	en-de, fr-en, ru-fr, en-fr, de-fr, ru-de, fr-de, ru-en, en-ru, fr-ru, de-ru, de-en	copy-source	ar-fr, ru-es, ur-en, fr-en, tr-en, zh-de, bg-en, ru-en, es-en, zh-en, sw-en, ja-ko, th-en, de-en, pl-mr, vi-en, hi-en, el-en, ar-en
addition omission	en-ca, en-el, en-et, en-ta, pl-en, hr-en, he-en, pl-sk, en-ar, ru-en, en-fi, zh-en, hu-en, be-en, lv-hr, en-he, ko-en, en-fa, sl-en, ca-en, en-gl, en-tr, en-sk, de-en, en-sr, fa-af, fa-en, ar-en, cs-en, en-de, en-hy, ar-hi, no-en, uk-en, fi-en, en-be, sr-pt, en-ru, sv-en, nl-en, sk-pl, en-hi, en-hu, mr-en, hi-ar, id-en, gl-en, en-fr, en-lv, fr-de, ca-es, en-uk,	addition omission	en-ur, en-hr, ur-en, en-no, en-sl, ro-en, en-vi, en-lt, es-en, en-nl, he-sv, en-it, en-ro, af-fa, en-id, lt-bg, en-af, af-en, es-ca, vi-en, sv-he, de-fr, pt-en, en-pl, et-en, hr-lv, wo-en, da-en, en-ko, en-da, ja-en, hy-en, pt-sr, hy-vi, fr-en, en-cs, lt-en, en-sv, tr-en, bg-en, lv-en, bg-lt, sr-en, en-es, en-bg, en-pt, hi-en, el-en, it-en

Table 11: Collection of list of languages per phenomena

Linguistically Motivated Evaluation of Machine Translation Metrics based on a Challenge Set

Eleftherios Avramidis and Vivien Macketanz
German Research Center for Artificial Intelligence (DFKI),
Speech and Language Technology, Berlin, Germany
firstname.lastname@dfki.de

Abstract

We employ a linguistically motivated challenge set in order to evaluate the state-of-the-art machine translation metrics submitted to the Metrics Shared Task of the 7th Conference for Machine Translation. The challenge set includes about 20,000 items extracted from 145 MT systems for two language directions (German⇔English), covering more than 100 linguistically-motivated phenomena organized in 14 categories. The best performing metrics are YiSi-1, BERTScore and COMET-22 for German-English, and UniTE, UniTE-ref, MetricX-XL-DA19 and MetricX-XXL-DA19 for English-German. Metrics in both directions are performing worst when it comes to named-entities & terminology and particularly measuring units. Particularly in German-English they are weak at detecting issues at punctuation, polar questions, relative clauses, dates and idioms. In English-German, they perform worst at present progressive of transitive verbs, future II progressive of intransitive verbs, simple present perfect of ditransitive verbs and focus particles.

1 Introduction

Automatic evaluation metrics have been valuable tools for Machine Translation (MT), allowing quick evaluation and suggesting directions for further development. Many metrics have been suggested throughout the years, which in turn sets the requirement for their evaluation.

Whereas MT metrics so far have been evaluated based on the agreement of their scores with human judgments on test sets drawn from broad text, little research has taken place on investigating whether the performance of the metrics generalizes enough when evaluating particular cases. A more target way of evaluating metrics is using *challenge sets*. These are targeted test sets, which have been devised in such a way, so that they benchmark the ability of metrics to score particular translation phenomena.

In this paper we present empirical results on the performance of MT metrics, using an extensive challenge set, which includes thousands of test items aiming to test the performance over more than one hundred linguistically-motivated phenomena in two language directions. It is based on thousands of manually created test items, their translation outputs from dozens of MT systems and semi-automatically evaluated with the supervision of linguists. Through this analysis we attempt to reveal strengths and weaknesses of several state-of-the-art MT metrics considering their background methods with regards to linguistic aspects.

The rest of the paper is structured as follows. In Section 2 related work is briefly described. In Section 3 we describe the construction of the challenge set and the evaluation protocol. The empirical results are outlined in Section 4, followed by a conclusion is Section 5.

2 Related work

The need for a thorough evaluation of Natural Language Processing (NLP) tools has lately received increased interest in the research community, indicated by a big amount of publications, among them several which received best paper awards (Ribeiro et al., 2020; Avelino et al., 2022; Campolungo et al., 2022). When focusing on MT, first efforts were made in the 1990s with the introduction of test suites (King and Falkedal, 1990), which were revived after the latest advances in the field (Guillou and Hardmeier, 2016). To the best of our knowledge, the first efforts relevant to the application of challenge sets on MT metrics was presented as an analysis at the Findings paper of the Metrics shared task of the 6th Conference of Machine Translation (Freitag et al., 2021), based on our test suite (Macketanz et al., 2022) that we are using on this paper.

Hereby we are advancing as to that preliminary analysis by (a) increasing the number of challenge

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 514 - 529
December 7-8, 2022 ©2022 Association for Computational Linguistics

items to about 9,000-10,000, including outputs from state-of-the-art systems from 2021, (b) adding a second language direction (English-German) (c) presenting a more fine-grained analysis, not only in the category level but also on the phenomenon level. This way we can get more confident and more generalisable empirical conclusions.

3 Method

3.1 Test suite for MT systems

The challenge set is based on our test suite (Macketanz et al., 2022), a manually devised test suite for MT for German-English and its recently developed extension for English-German (Macketanz et al., 2021).[1] The German-English side consists of 5,540 German test sentences covering 107 linguistically motivated phenomena, organized in 14 categories. The English-German side consists of 4,438 English test sentences covering 105 phenomena, organized in 12 categories.

The chosen phenomena do not follow a particular linguistic theory but their definition has been inspired by observing linguistic aspects which are relevant for MT. Each phenomenon is represented by at least 20 source test sentences to guarantee a balanced test set. The test suite is used to evaluate MT systems with regard to their performance on the phenonmenon-targeting test sentences. The evaluation operates semi-automatically and it occurs based on a set of handwritten rules which contain regular expressions and fixed string tokens.

The above described test suite has been used to evaluate the outputs of 116 German-English and 29 English-German systems, submitted at the translation task of the Conference of Machine Translation (WMT) for four consequent years (2018-2021; Macketanz et al., 2018; Avramidis et al., 2019, 2020; Macketanz et al., 2021), including a preliminary system comparison in 2017 (Burchardt et al., 2017).

3.2 Challenge set for MT metrics

Here we describe how the aforementioned test suite, along with inputs from previous shared tasks, is used in order to evaluate MT metrics. A challenge set for metrics requires contrastive pairs of correct/incorrect translations and a reference, whereas our original test suite contained only source sentences and handwritten rules for the outputs, but no reference translations. We therefore use the collected MT outputs to construct the challenge items for the metrics task in order to create the required challenge sets as following. For every source sentence of the test suite we create a tuple including:

- one correct translation, to be given to the metrics as reference translation; and a pair of
- another correct translation and
- one incorrect translation, the latter two intended to be given to the metrics for scoring.

In order to generate these tuples we perform random combinations of correct and wrong translations from the WMT outputs. Also, before collecting MT outputs, we filter out a part of the original test items, to be reserved for future evaluations.

The above process resulted into a metrics challenge set with 10,402 items for German-English and 8,945 items for English-German. The fact that the correct and incorrect translations have been sampled from real MT system outputs of the last 4 years, implies that these challenge set is closer to the real MT system ecosystem, as compared to artificially created challenge sets, which may contain translations that would never be produced by state-of-the-art MT.

3.3 Evaluation of metrics

As explained, the challenge set consists of subsets of challenge items, where every subset has been deliberately created so that it can detect the metrics' performance to a particular phenomenon. For every challenge item, the two MT outputs (correct/incorrect) are given unlabelled to the metrics as two separate MT hypotheses so that they score them against the aforementioned references and/or the source. The item is considered correctly scored, if the metric gives to the correct MT output a higher score than the incorrect MT output. Then the following statistics are calculated:

Accuracy per phenomenon is given by the ratio of all correctly-scored challenge items per phenomenon to the total number of challenge items for this phenomenon

Accuracy per category is given by the ratio of all correctly-scored challenge items per category to the total number of challenge items for this category (after aggregating the underlying phenomena of this category in one set).

Significant tests for comparisons: the highest metric accuracy for every phenomenon is compared to all other metric accuracies of the same

[1] https://github.com/DFKI-NLP/mt-testsuite

phenomenon. For this, a one-tailed Z-test with $\alpha = 0.95$ is calculated. The metrics whose accuracies that are not significantly worse than the highest accuracy, are considered to share the winning position for this phenomenon. The best accuracies per category are calculated in the same way, after aggregating the challenge items from the underlying phenomena of every category.

Statistics for metric categories: We repeat this significance testing in two levels: one for all metrics participating in the shared task, and then separately for each one of the three metric categories (baseline, QE as a metric, reference-based). The significantly best systems per phenomenon over all metrics are indicated with a gray background, whereas the significantly best systems per metrics category are indicated with boldface.

Finally, we report three kinds of average scores: **Micro-average** treats all items equally, aggregating all test items to compute the average percentages; **Category macro-average** treats all categories equally by computing the percentages independently for each category and then averaging them **Phenomenon macro-average** treats all phenomena equally, by computing the percentages independently for each phenomenon and then averaging them

4 Results

The results are displayed in detail in Tables 1 and 3 in the category level and in Tables 4 and 5 for the phenomenon level, for both language directions German-English and English-German respectively.

4.1 Metric performance analysis

Here we are observing the statistics with a focus on comparing the performance of various metrics on the challenge set.

German-English The best performing metrics for German-English are YiSi-1 (Lo, 2019), BERTScore (Zhang et al., 2020) and COMET-22 (Rei et al., 2022), achieving the significantly highest micro- and macro-average accuracies (84-85%), whereas for the macro-average, UniTE-ref (Wan et al., 2022) is also included in the first significance cluster. The two QE based metrics of HWTSC (Liu et al., 2022) get the lowest accuracies, together with the baseline BLEU (Papineni et al., 2002).

When considering the systems performance with regards to particular categories, one can see that different metrics win in different combinations of categories. Most reference-based metrics perform best for at least four categories, apart from MS-COMET which only gets two.

Interestingly enough, one QE method is outperforming reference-based metrics for one category: HWTSC-TLM is the best performing system for *punctuation*. Additionally, UNITE-src performs equally well to reference-based metrics for coordination and ellipsis.

English-German UniTE and UniTE-ref are the winning metrics based on the macro-average (82%), whereas the former seems to be stronger than the latter, winning 5 categories. MetricX-XL-DA19 and MetricX-xxl-DA19 are the winning metrics when it comes to micro-average (78%). Their average accuracies are close to 80%, which raises concerns, as this indicates that 2 out of 10 challenge items in average are not scored correctly in this language direction, even for the best performing metrics. The lowest scoring metric is MATESE (Perrella et al., 2022) in both QE and reference-based versions, very close to REUSE (Mukherjee and Shrivastava, 2022).

Also in this direction, QE methods manage to outperform submitted reference-based metrics in a few categories. REUSE is the best performing metric for *false friends* and UNITE-src for *function words*. COMET-kiwi (Rei et al., 2022) and UniTE-src are on par with reference-aware metrics when it comes to *subordination* and Cross-QE (Liu et al., 2022) for *verb tense/aspect/mood*.

4.2 Linguistically motivated analysis

Here we are looking closer to the results for particular phenomena or categories.

4.2.1 German-English

Category-level The overall average accuracy of all metrics with regards to the linguistically motivated categories is at 78% for German-English. This indicates that the metrics failed in average to predict properly the scores for about one out of four challenge items that we provided. Even for the best categories, the accuracy achieved by most metrics is considerably below the acceptable limit of 90%.

The best performing category in *negation* with 86% average accuracy. For the rest of the categories, the average accuracy is less than 82%. The worst performing categories in average are *named entity and terminology* and *punctuation* with only 67% accuracy, whereas *subordination* comes next

		baselines								QE as a metric							ref. based metrics								
ling. category	#	BERTScore	BLEU	BLEURT-20	COMET-20	YiSi-1	chrF	f101spBLEU	f200spBLEU	COMETKiwi	Cross-QE	HWTSC-TLM	HWTSC-TS	KG-BERT	MS-COMET-QE	UniTE-src	COMET-22	MS-COMET	UniTE-ref	UniTE	XL-DA19	XL-MQM20	XXL-DA19	XXL-MQM20	avg
Ambiguity	298	**90**	71	88	86	89	80	81	79	**82**	73	60	65	67	**82**	80	87	85	**89**	89	88	**90**	83	86	81
Composition	252	88	65	87	85	**90**	74	70	71	76	**77**	73	76	59	72	75	83	86	82	83	86	82	**87**	82	79
Coordination & ellipsis	316	**79**	74	**79**	77	**80**	77	72	73	**82**	78	69	72	78	69	**83**	**84**	75	79	80	79	**83**	78	78	77
False friends	90	91	64	**93**	82	**92**	78	69	70	88	74	81	**91**	87	63	44	**91**	88	**92**	**92**	90	90	87	88	82
Function word	586	**83**	72	**83**	78	81	73	73	73	**81**	77	78	**81**	70	68	77	83	81	**86**	84	84	79	83	82	79
LDD & interrogatives	1014	**85**	75	**84**	**85**	**85**	76	74	74	**84**	**83**	72	75	63	81	**82**	**86**	83	84	85	**85**	82	**85**	82	80
MWE	610	**85**	73	**85**	**85**	**86**	78	74	75	**76**	**76**	70	60	56	60	73	86	82	**89**	**90**	88	88	87	81	78
Named entity & termin.	861	74	62	68	68	**76**	67	70	71	65	**71**	64	61	55	61	61	70	66	67	64	67	**75**	70	72	67
Negation	76	**95**	84	88	92	91	88	83	80	**93**	78	62	74	87	88	92	91	88	**93**	**93**	89	78	88	83	86
Non-verbal agreement	419	77	74	**83**	81	76	75	75	76	75	72	66	63	62	**78**	73	**84**	77	84	**85**	83	81	**85**	83	77
Punctuation	293	74	77	70	68	73	69	78	**80**	55	75	**81**	73	62	61	69	**68**	65	65	61	61	53	59	47	67
Subordination	679	76	69	**77**	**77**	74	69	68	69	72	**75**	59	62	65	64	73	**80**	77	77	78	75	70	78	74	72
Verb tense/aspect/mood	4697	**88**	69	85	86	**89**	77	71	71	81	**87**	63	71	78	81	82	**86**	83	**85**	**85**	84	79	**85**	81	80
Verb valency	211	**91**	70	88	88	**90**	72	69	69	**86**	72	64	64	62	75	82	**94**	88	91	91	91	88	91	88	81
macro avg.	10402	**84**	71	83	81	**84**	75	73	74	**78**	76	69	70	68	72	75	**84**	80	**83**	**83**	82	80	82	79	78
micro avg.	10402	**84**	70	82	82	**85**	75	72	72	78	**81**	66	70	70	75	78	**84**	80	**83**	82	82	79	82	79	78

Table 1: Accuracy of the metrics (%) with regards to the 14 linguistically motivated categories for German-English. The significantly best systems per phenomenon over all metrics are indicated with a gray background, whereas the significantly best systems per metrics category are indicated with boldface.

with 72%. The lowest performing score for all systems and all categories is achieved by MetricX-XL-MQM20, which can only score correctly almost half of the punctuation challenge items (53%).

Phenomenon-level The best accuracy for this language pair is achieved for *Transitive, future I* where the metrics get an accuracy of 95%-100%. Another 13 phenomena score more than 85%. Four of them also refer to the future tenses of the transitive, in particular future I and future II in both the plain and their subjunctive form. Additionally, one can see good performance in *Intransitive-present, Modal-future I, pied-piping, comma, negation, passive voice,* and the *negated modal for future I subjunctive II.*

The lowest accuracy of all metrics in average is given for *polar questions* (61%), followed by *quotation marks* (63%). An average accuracy of less than 65% is given for some more phenomena, such as the ones including *measuring units, relative clauses, dates* and *idioms*.

The lowest phenomenon accuracies are given by QE methods, and particularly when it comes to *idioms*, where HWTSC-TLM achieves the lowest performance of 17%. This is explainable by the fact that idioms require resolving rather rare semantic relations between the source and the MT output (used for QE), but can be easily resolved with lexical matching on the reference (used by reference-aware metrics). Idioms have shown to be a particular challenge for MT systems as well.

4.2.2 English-German

Category-level The overall average accuracy of all metrics (Table 3) with regards to the linguistically motivated categories is at 69-72% for English-German. This is 6% lower than the respective average accuracy for German-English, indicating that the metrics for this MT language direction perform worse.

The category where all metrics perform best in average is *negation* (86%), whereas the one where they perform worse is *Named entity & terminology* (59%). The rest of the categories lie in rather mediocre accuracies, between 66% and 82%. The performance of metrics in English-German is worse than German-English in all categories apart from *function words, punctuation* and *subordination*, although the comparisons between the language directions have to be taken with a grain of salt, due to the fact that the two directions consist of different items.

Phenomenon-level The English-German phenomena, where metrics perform best in average are

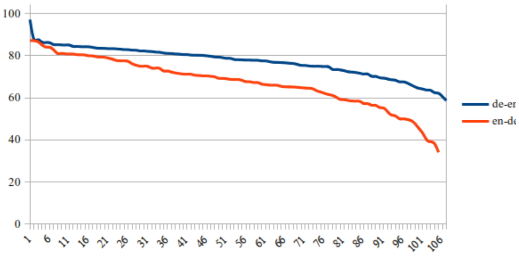


Figure 1: Plot of the accuracy of all phenomena per language direction. The accuracy percentage is shown on the vertical axis and the phenomena on the horizontal

the *Contact clause, Negation, Ditransitive - present progressive* and *question tags*, achieving more than 85% of accuracy. The most difficult phenomena to score are the *Intransitive - future II progressive* and the *Transitive - present progressive*, as they achieve less then 40% average accuracy, followed by *Ditransitive - present perfect simple*, *measuring units* and *focus particles*.

Interestingly enough, in this language direction there are metrics which scored zero accuracies in several phenomena, something that we didn't see in the opposite language direction.[2] These zero accuracies are mostly relevant to rare verb-related phenomena (e.g. intransitive constructions). A comparative plot of the accuracies for all phenomena for both language directions can be seen in Figure 1. It is very clear that English-German lacks considerably, with its lowest scored phenomena having an accuracy at half of the lower-scored phenomena of the opposite direction.

Finally, some examples of incorrectly scored challenge items from the phenomena that have the lowest accuracies can be seen in Table 2. Whereas is hard to know why each metric score in a wrong way, in many cases we may assume that it was misled by a part of the sentence which seemed distant to reference (or the source for QE), but it was correct.

5 Conclusion

In this paper we analyzed the performance of several state-of-the-art metrics with regards to particular linguistically-motivated phenomena for two language pairs, German-English and English-German. The analysis gave a multitude of observations, regarding both the performance of the metrics and the corresponding linguistic observations.

In an effort to draw conclusions after averaging accuracies, we conclude that the best performing metrics are YiSi-1, BERTScore and COMET-22 for German-English, and UniTE, UniTE-ref, MetricX-XL-DA19 and MetricX-xxl-DA19 for English-German.

The metrics are particularly good at scoring the German-English verb tense *Transitive, future I* and the category of *negation*. Concerning English-German, the best performing phenomena are *contact clause* and *negation*.

On the contrary, metrics in both directions are performing worst when it comes to *named-entities & terminology*. Particularly in German-English they are weak at detecting issues at *punctuation (quotation marks), polar questions, measuring units, relative clauses, dates* and *idioms*. In English-German at *present progressive of transitive verbs*, *future II progressive of intransitive verbs*, *present perfect of ditransitive verbs, measuring units* and *focus particles*.

We believe that further investigation on particular phenomena or categories can provide explanations for the relevant observations and possibly lead to suggestions for technical improvements in the development of the metrics in the future. For example, many observations are also relevant to whether the metrics take into account for scoring the reference translation or the source (QE as a metric). Additionally, having seen several low accuracies regarding punctuation, we note that this issue is often handled via pre-processing scripts. The low percentages of scoring punctuation issues, show that the metrics should improve their engineering on that direction.

Acknowledgements

This research was supported by the Deutsche Forschungsgemeinschaft (DFG) through the project TextQ and by the German Federal Ministry of Education through the project SocialWear (grant num. d01IW20002). We would like to thank Hans Uszkoreit, Aljoscha Burchardt, Ursula Strohriegel, Renlong Ai and He Wang for their prior contributions for the creation of the test suite.

References

Mariana Avelino, Vivien Macketanz, Eleftherios Avramidis, and Sebastian Möller. 2022. A Test Suite

[2] again this should take into consideration that English-German set has a participation of less systems and therefore less diversity than German-English

for the Evaluation of Portuguese-English Machine Translation. In *Computational Processing of the Portuguese Language*, pages 15–25, Cham. Springer International Publishing.

Eleftherios Avramidis, Vivien Macketanz, Ursula Strohriegel, Aljoscha Burchardt, and Sebastian Möller. 2020. Fine-grained linguistic evaluation for state-of-the-art Machine Translation. In *Proceedings of the Fifth Conference on Machine Translation*. Association for Computational Linguistics.

Eleftherios Avramidis, Vivien Macketanz, Ursula Strohriegel, and Hans Uszkoreit. 2019. Linguistic evaluation of German-English Machine Translation using a Test Suite. In *Proceedings of the Fourth Conference on Machine Translation. Conference on Machine Translation (WMT-2019)*, pages 644–653, Florence, Italy. Association for Computational Linguistics.

Aljoscha Burchardt, Vivien Macketanz, Jon Dehdari, Georg Heigold, Jan-Thorsten Peter, and Philip Williams. 2017. A Linguistic Evaluation of Rule-Based, Phrase-Based, and Neural MT Engines. *The Prague Bulletin of Mathematical Linguistics*, 108:159–170.

Niccolò Campolungo, Federico Martelli, Francesco Saina, and Roberto Navigli. 2022. DiBiMT: A novel benchmark for measuring Word Sense Disambiguation biases in Machine Translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4331–4352, Dublin, Ireland. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Liane Guillou and Christian Hardmeier. 2016. PROTEST: A test suite for evaluating pronouns in machine translation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 636–643, Portorož, Slovenia. European Language Resources Association (ELRA).

Margaret King and Kirsten Falkedal. 1990. Using test suites in evaluation of machine translation systems. In *Proceedings of the 13th conference on Computational Linguistics*, volume 2, pages 211–216, Morristown, NJ, USA. Association for Computational Linguistics.

Yilun Liu, Xiaosong Qiao, Zhanglin Wu, Su Chang, Min Zhang, Yanqing Zhao, shimin tao Song Peng, Hao Yang, Ying Qin, Jiaxin Guo, Minghan Wang, Yinglu Li, Peng Li, and Xiaofeng Zhao. 2022. Partial Could Be Better Than Whole: HW-TSC 2022 Submission for the Metrics Shared Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Vivien Macketanz, Eleftherios Avramidis, Aljoscha Burchardt, and Hans Uszkoreit. 2018. Fine-grained evaluation of German-English Machine Translation based on a Test Suite. In *Proceedings of the Third Conference on Machine Translation (WMT18)*, pages 578–587, Brussels, Belgium. Association for Computational Linguistics.

Vivien Macketanz, Eleftherios Avramidis, Aljoscha Burchardt, He Wang, Renlong Ai, Shushen Manakhimova, Ursula Strohriegel, Sebastian Möller, and Hans Uszkoreit. 2022. A linguistically motivated test suite to semi-automatically evaluate German–English machine translation output. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 936–947, Marseille, France. European Language Resources Association.

Vivien Macketanz, Eleftherios Avramidis, Shushen Manakhimova, and Sebastian Möller. 2021. Linguistic evaluation for the 2021 state-of-the-art Machine Translation systems for German to English and English to German. In *Proceedings of the Sixth Conference on Machine Translation. (WMT21)*, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ananya Mukherjee and Manish Shrivastava. 2022. REUSE: REference-free UnSupervised quality Estimation Metric. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Stefano Perrella, Lorenzo Proietti, Alessandro Scirè, Niccolò Campolungo, and Roberto Navigli. 2022. Machine Translation Evaluation as a Sequence Tagging Problem. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 Submission for the Metrics Shared Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Yu Wan, Keqin Bao, Dayiheng Liu, Baosong Yang, Derek F. Wong, Lidia S. Chao, Wenqiang Lei, and Jun Xie. 2022. Alibaba-Translate China's Submission for WMT2022 Metrics Shared Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Appendix

		German-English
idiom	src	Ich glaube, Tim hat ein Auge auf Lena geworfen.
	ref	I think Tim has a crush on Lena.
	✓	I think Tim has cast an eye on Lena.
	✗	I think Tim has an eye on Lena.
polar question	src	Willst du mit mir ins Kino gehen?
	ref	Do you want to go to a movie with me?
	✓	Do you want to go with me into the cinema?
	✗	You want to go to the cinema with me?
measuring unit	src	Ein ausgewachsener Afrikanischer Elefant wiegt etwa sechs Tonnen.
	ref	An adult African elephant weighs about six tons.
	✓	A fully grown African elephant weighs about six tons.
	✗	An adult African elephant weighs about six tonnes.
comma	src	Er fragte sich, welches Auto er kaufen sollte.
	ref	He wondered what car to buy.
	✓	He wondered which car to buy.
	✗	He asked himself, which car he should buy.
quotation marks	src	"Wann sollen wir uns treffen?", wollten sie wissen.
	ref	"When are we supposed to meet?" they asked.
	✓	"When shall we meet?" they wanted to know.
	✗	When are we going to meet? They wanted to know.
		English-German
Intransitive . future II progr	src	They will have been running.
	ref	Sie werden gelaufen sein.
	✓	Sie werden gerannt sein.
	✗	Sie würden gelaufen sein.
Focus particle	src	He even drank four bottles of wine.
	ref	Er habe sogar vier Flaschen Wein getrunken.
	✓	Er trank sogar vier Flaschen Wein.
	✗	Er trank noch vier Flaschen Wein.
Transitive present progr.	src	They are playing the piano.
	ref	Sie spielen auf dem Klavier.
	✓	Sie spielen Klavier.
	✗	Sie spielen das Klavier.
measuring unit	src	Potatoes are sold in hundredweights.
	ref	Kartoffeln werden in Zentnergewichten verkauft.
	✓	Kartoffeln werden in Zentner verkauft.
	✗	Kartoffeln werden in Hundertgewichten verkauft.

Table 2: Indicative examples of incorrectly scored challenge items for the phenomena that have the lowest accuracies

		baselines								QE as a metric									ref. based metrics												
ling. category	#	BERTScore	BLEU	BLEURT-20	COMET-20	YiSi-1	chrF	f101spBLEU	f200spBLEU	COMETKiwi	Cross-QE	HWTSC-TLM	HWTSC-TS	KG-BERT	MATESE-QE	MS-COMET-QE	REUSE	UniTE-src	COMET-22	MATESE	MEE	MEE2	MEE4	MS-COMET	UniTE-ref	UniTE	XL-DA19	XL-MQM20	XXL-DA19	XXL-MQM20	avg
Ambiguity	146	87	71	**90**	82	87	**89**	87	88	55	47	**81**	47	47	25	38	15	36	84	40	73	88	91	78	**97**	93	94	88	95	87	72
Coordination & ellipsis	836	69	61	**80**	76	73	61	64	62	**76**	71	72	70	60	33	70	38	74	79	37	59	62	62	78	79	78	81	**83**	81	80	68
False friends	225	66	63	70	**73**	67	72	66	67	67	60	64	73	68	52	73	**89**	64	69	35	69	79	**88**	76	71	71	71	71	68	69	69
Function word	200	90	76	90	**94**	78	72	74	73	91	92	78	90	90	66	92	66	**94**	**90**	28	78	80	85	90	**90**	**91**	78	82	84	82	82
MWE	829	79	72	**87**	82	85	77	74	73	78	**81**	79	**82**	**82**	37	71	32	78	**86**	46	69	76	78	81	**87**	**87**	86	**86**	79	77	75
Named entity & termin.	1272	58	55	**66**	63	**64**	61	63	64	55	**59**	56	53	54	21	55	43	53	61	30	59	63	63	62	69	68	**73**	70	**73**	**72**	59
Negation	174	87	83	89	90	**93**	85	82	84	**92**	86	87	**91**	**91**	43	**92**	78	90	91	79	84	92	92	90	**94**	**94**	82	81	82	78	86
Non-verbal agreement	372	75	72	81	**84**	78	70	74	75	77	70	59	63	59	34	**79**	39	72	**90**	48	61	73	76	84	87	86	88	**90**	**90**	**90**	73
Punctuation	336	70	**79**	76	77	77	74	71	68	68	72	70	51	51	50	68	46	**79**	79	51	64	75	74	73	**81**	**81**	67	60	72	68	69
Subordination	994	77	74	80	**83**	78	74	75	73	**86**	82	81	84	82	47	83	48	**85**	**84**	53	73	77	78	82	**85**	**85**	**85**	84	82	79	77
Verb tense/aspect/mood	3081	67	62	**70**	69	**69**	69	64	64	70	**77**	51	58	59	41	61	54	70	**77**	43	71	71	69	64	70	72	**78**	74	76	73	66
Verb valency	480	73	64	**84**	74	76	71	66	70	**82**	74	65	69	68	30	70	48	72	82	42	62	70	70	76	79	80	79	78	**85**	81	70
macro avg.	8945	75	69	**80**	**79**	77	73	72	72	**75**	73	70	69	68	40	71	50	72	81	44	69	75	77	78	**82**	**82**	80	79	80	78	72
micro avg.	8945	70	65	**76**	74	73	69	68	68	**73**	**74**	63	65	64	38	67	48	71	78	42	68	71	72	72	77	77	**79**	77	**78**	76	69

Table 3: Accuracy of the metrics (%) with regards to the 12 linguistically motivated categories for English-German

			baselines								QE as a metric							ref. based metrics								
ling. category	ling. phenomenon	#	BERTScore	BLEU	BLEURT-20	COMET-20	YiSi-1	chrF	f101spBLEU	f200spBLEU	COMETKiwi	Cross-QE	HWTSC-TLM	HWTSC-TS	KG-BERT	MS-COMET-QE	UniTE-src	COMET-22	MS-COMET	UniTE-ref	UniTE	XL-DA19	XL-MQM20	XXL-DA19	XXL-MQM20	avg
Ambiguity	Lexical ambiguity	129	91	74	**95**	**94**	88	87	82	82	81	65	56	60	57	**82**	**83**	93	81	**97**	**97**	95	93	89	88	83
	Structural ambiguity	169	**89**	69	83	80	**89**	75	80	76	**82**	79	64	69	75	**82**	78	83	**88**	83	82	82	**88**	78	84	80
Composition	Compound	129	86	64	**90**	83	**91**	74	68	70	**71**	69	64	**70**	45	64	67	81	87	82	83	90	88	**93**	88	77
	Phrasal verb	123	**91**	66	85	86	89	74	72	72	81	**86**	83	82	74	80	85	**84**	**85**	82	82	81	76	81	75	80
Coordination & ellipsis	Gapping	51	71	76	**82**	78	71	76	73	75	**100**	98	59	75	75	84	88	98	86	94	90	80	**100**	88	94	83
	Right node raising	67	90	70	76	75	**91**	75	70	67	78	**84**	64	55	82	72	72	**82**	75	78	76	76	79	75	78	76

Table 4: Accuracy of the metrics(%) with regards to the linguistically-motivated phenomena for German-English

(Continued on next page)

			baselines								QE as a metric							ref. based metrics								
ling. category	ling. phenomenon	#	BERTScore	BLEU	BLEURT-20	COMET-20	YiSi-1	chrF	f101spBLEU	f200spBLEU	COMETKiwi	Cross-QE	HWTSC-TLM	HWTSC-TS	KG-BERT	MS-COMET-QE	UniTE-src	COMET-22	MS-COMET	UniTE-ref	UniTE	XL-DA	XL-MQM	xxl-DA19	xxl-MQM20	avg
	Sluicing	128	**80**	75	77	77	78	**79**	73	75	80	66	77	79	76	59	**86**	**80**	71	73	78	**81**	76	79	70	76
	Stripping	70	76	74	**84**	79	80	76	73	73	77	80	67	71	**83**	73	**83**	81	76	81	80	77	**89**	71	81	78
False friends	False friends	90	91	64	**93**	82	**92**	78	69	70	88	74	81	**91**	87	63	44	**91**	88	**92**	**92**	90	90	87	88	82
Function word	Focus particle	64	86	75	83	**89**	88	75	72	77	83	70	70	84	**88**	81	75	84	86	86	**88**	**88**	67	**88**	81	81
	Modal particle	166	**87**	79	85	83	**86**	77	80	81	**82**	75	69	81	**83**	67	**83**	**89**	82	**89**	**88**	**89**	83	87	83	82
	Question tag	356	80	69	**82**	74	78	71	69	69	81	79	**84**	80	61	65	75	81	79	**84**	81	81	79	81	81	77
LDD & interrogatives	Extended adjective construction	320	**87**	80	**88**	87	**88**	80	80	80	90	**93**	79	82	61	91	88	**90**	87	88	**89**	**89**	86	88	84	85
	Extraposition	92	73	74	75	**82**	77	**83**	72	73	67	74	65	**79**	62	63	75	76	74	67	77	80	79	**84**	78	74
	Multiple connectors	87	74	**79**	63	72	76	76	**80**	**79**	**70**	68	67	63	64	**69**	**70**	68	**79**	61	63	66	57	66	53	69
	Pied-piping	162	94	78	93	**96**	93	77	75	75	**96**	90	73	74	70	79	94	**95**	**94**	**94**	**94**	**94**	89	**94**	90	87
	Polar question	51	**71**	43	63	61	67	45	45	47	69	49	49	53	61	69	**78**	67	55	65	71	61	**75**	61	**75**	61
	Scrambling	144	**90**	72	**90**	87	88	74	69	69	**90**	88	82	81	51	**90**	81	**98**	90	93	90	96	92	95	95	85
	Topicalization	61	85	85	**87**	84	**87**	84	**87**	**87**	77	69	66	70	77	**82**	74	82	74	80	80	82	70	**85**	79	80
	Wh-movement	97	79	62	**85**	81	77	69	63	63	72	**75**	56	64	66	**75**	**74**	81	73	**86**	84	80	73	78	75	74
MWE	Collocation	190	87	72	**91**	89	88	79	74	74	**84**	82	82	65	67	73	79	89	79	92	**93**	90	91	91	89	83
	Idiom	133	82	67	76	**85**	83	69	67	65	44	**55**	36	17	20	31	33	75	77	87	**88**	**89**	86	86	75	65
	Prepositional MWE	146	84	79	**85**	84	**86**	84	79	81	82	**84**	82	84	72	71	**85**	85	78	84	**86**	**86**	**86**	85	77	82
	Verbal MWE	141	86	74	**87**	80	84	77	77	77	89	81	77	68	57	60	**91**	92	**95**	93	91	87	87	84	82	82
Named entity & termin.	Date	203	**67**	50	65	65	66	58	58	57	**70**	**70**	63	68	68	61	66	67	63	67	63	69	**74**	68	72	65
	Domainspecific term	214	71	63	71	64	**74**	71	68	68	67	**77**	63	57	59	66	60	72	64	72	71	68	**75**	71	70	68
	Location	181	78	65	70	75	**82**	66	71	74	62	57	**76**	64	38	56	54	75	71	66	61	68	**80**	70	78	68
	Measuring unit	203	75	67	61	64	77	72	81	**84**	57	**73**	54	51	56	56	55	63	62	59	55	62	**67**	66	66	64
	Proper name	60	90	75	85	87	**92**	73	77	77	78	**88**	72	70	50	70	83	85	**90**	83	83	78	85	**90**	88	80
Negation	Negation	76	**95**	84	88	92	91	88	83	80	**93**	78	62	74	87	88	92	91	88	**93**	**93**	89	78	88	83	86
Non-verbal agreement	Coreference	251	74	68	**90**	85	75	72	71	71	81	77	73	69	66	**84**	78	**91**	82	90	90	**91**	88	**92**	**91**	80
	External possessor	104	84	**88**	75	76	82	**88**	85	86	70	68	50	51	58	68	**74**	**76**	73	75	**77**	71	71	75	70	73
	Internal possessor	64	80	80	72	72	72	67	78	**83**	61	59	62	58	52	**67**	53	69	61	73	**77**	72	69	72	72	69
Punctuation	Comma	46	91	91	**93**	85	89	87	91	91	85	**91**	83	85	87	80	80	89	85	87	83	89	80	**91**	83	87
	Quotation marks	247	71	75	66	64	70	65	76	**77**	49	72	**81**	71	57	57	67	**64**	61	60	57	56	48	53	40	63
Subordination	Adverbial clause	87	71	70	**82**	75	72	67	66	67	70	**74**	66	68	70	69	66	**77**	67	74	72	70	74	75	68	71
	Cleft sentence	109	**73**	**73**	67	71	66	66	70	69	66	69	48	64	62	55	**71**	72	**75**	66	69	66	64	65	61	66
	Free relative clause	70	63	67	**77**	71	67	71	74	71	60	70	50	56	63	69	**77**	77	**83**	80	81	74	54	76	67	70
	Indirect speech	119	76	64	**81**	**80**	71	70	62	63	**80**	75	58	58	57	62	70	**87**	**87**	**86**	**86**	83	65	84	82	73
	Infinitive clause	64	**78**	77	77	72	**78**	77	75	73	**73**	70	62	67	**73**	72	70	70	67	73	70	75	66	**80**	67	72
	Object clause	54	85	74	85	**91**	89	81	72	72	76	89	69	69	**94**	67	80	**93**	87	91	91	89	80	87	85	82

Table 4: Accuracy of the metrics(%) with regards to the linguistically-motivated phenomena for German-English

(Continued on next page)

			baselines								QE as a metric							ref. based metrics								
ling. category	ling. phenomenon	#	BERTScore	BLEU	BLEURT-20	COMET-20	YiSi-1	chrF	f101spBLEU	f200spBLEU	COMETKiwi	Cross-QE	HWTSC-TLM	HWTSC-TS	KG-BERT	MS-COMET-QE	UniTE-src	COMET-22	MS-COMET	UniTE-ref	UniTE	XL-DA	XL-MQM	xxl-DA19	xxl-MQM20	avg
	Pseudo-cleft sentence	25	**96**	72	68	88	92	60	72	72	80	**100**	48	32	60	72	96	88	80	**92**	**92**	80	88	88	68	78
	Relative clause	71	**70**	63	65	**70**	66	66	63	68	63	61	59	59	48	51	**66**	**77**	65	63	70	62	66	73	73	65
	Subject clause	80	**85**	66	**85**	**86**	**86**	65	62	71	**86**	**86**	68	70	62	70	80	86	84	82	82	85	88	85	**92**	79
Verb tense/aspect/mood	Conditional	50	**80**	**80**	76	76	**80**	**80**	78	76	80	80	82	78	74	80	**90**	82	78	82	80	82	**88**	76	78	80
	Ditransitive - future I	121	87	72	**92**	89	88	71	70	70	**93**	**92**	58	68	85	89	91	**94**	80	**94**	**94**	92	85	92	92	84
	Ditransitive - future I subjunctive II	84	90	63	89	93	**95**	75	68	68	90	**94**	50	65	92	**94**	**93**	**92**	90	**92**	**92**	**93**	88	**93**	90	85
	Ditransitive - future II	97	**94**	60	82	73	**94**	71	67	67	**98**	**98**	58	69	69	96	93	**93**	66	88	85	85	78	88	80	80
	Ditransitive - future II subjunctive II	88	93	73	86	88	**97**	78	69	69	88	**99**	65	75	89	97	93	89	77	**92**	90	83	84	85	84	84
	Ditransitive - perfect	72	**93**	62	81	78	**93**	72	67	67	93	**96**	46	58	75	88	**96**	86	71	**88**	81	81	82	**88**	81	79
	Ditransitive - pluperfect	86	83	67	83	77	**88**	79	71	71	81	**83**	57	71	74	**84**	**83**	**86**	62	**86**	**86**	79	69	83	62	77
	Ditransitive - pluperfect subjunctive II	107	**94**	71	79	86	92	88	67	67	64	69	65	66	75	68	**92**	86	75	**90**	86	82	78	77	71	78
	Ditransitive - present	90	82	61	**91**	86	81	77	66	64	70	**99**	56	60	83	81	78	**88**	**89**	**89**	**89**	83	86	**89**	**89**	80
	Ditransitive - preterite	117	84	62	85	**88**	**89**	76	68	68	84	**91**	62	61	87	84	84	**95**	86	91	90	90	92	**94**	93	83
	Ditransitive - preterite subjunctive II	110	87	61	**95**	93	90	85	65	65	85	**87**	60	61	85	85	85	**96**	89	95	95	95	**96**	**97**	**97**	85
	Imperative	98	88	78	**95**	92	89	79	81	76	**88**	84	78	74	57	78	86	88	86	**92**	**92**	90	87	**91**	90	84
	Intransitive - future I	32	84	53	88	**91**	**91**	72	59	59	84	97	69	91	**100**	94	97	84	**94**	88	88	88	88	88	91	84
	Intransitive - future I subjunctive II	56	**93**	61	**93**	91	89	70	73	71	93	95	55	68	**100**	86	89	95	88	98	**100**	98	84	96	98	86
	Intransitive - future II	62	87	60	**90**	84	89	77	65	69	79	**87**	45	58	69	63	60	90	92	94	94	95	94	**97**	92	80
	Intransitive - future II subjunctive II	94	**97**	72	94	91	**98**	89	76	74	80	**100**	63	82	86	84	71	91	86	**94**	**93**	**94**	85	**93**	87	86
	Intransitive - perfect	61	85	56	84	72	**87**	59	56	54	62	**69**	66	66	64	59	59	72	79	69	70	75	67	**82**	72	69
	Intransitive - pluperfect	85	85	79	85	80	**87**	85	81	76	68	86	46	55	**88**	64	61	78	78	**81**	**81**	**82**	71	**81**	69	76
	Intransitive - pluperfect subjunctive II	79	**100**	87	97	96	**100**	90	80	80	78	**94**	56	76	**95**	73	71	**96**	91	**97**	**97**	**96**	95	**97**	95	89
	Intransitive - present	54	96	69	91	94	**98**	74	69	72	96	**98**	65	76	94	94	91	**94**	93	93	93	89	93	87	87	87
	Intransitive - preterite	46	70	46	**89**	78	74	63	46	52	**93**	**93**	74	83	91	85	85	87	76	85	80	85	80	**89**	74	77
	Intransitive - preterite subjunctive II	100	81	43	**86**	79	79	51	60	61	79	89	58	67	**91**	83	77	**88**	81	83	84	87	**89**	82	80	76
	Modal - future I	42	**98**	90	88	95	95	95	90	90	83	**98**	76	83	74	90	74	**88**	**88**	**88**	**88**	**88**	76	86	81	87
	Modal - future I subjunctive II	86	**97**	94	81	93	**97**	94	93	93	**79**	**78**	**78**	**79**	67	**78**	62	**85**	80	**85**	**86**	**85**	64	**86**	65	83
	Modal - perfect	149	**85**	72	74	79	**85**	72	74	74	**85**	81	67	77	47	60	72	70	**78**	67	65	66	57	70	61	71
	Modal - pluperfect	75	**100**	**100**	84	95	**100**	99	**100**	**100**	69	89	83	**91**	47	49	75	76	**85**	72	72	69	44	71	48	79
	Modal - pluperfect subjunctive II	61	87	72	79	89	**90**	80	69	69	85	**90**	69	79	85	87	87	84	**87**	84	80	80	74	82	77	81
	Modal - present	30	83	57	**93**	87	80	73	63	63	**90**	80	53	80	80	83	83	80	80	80	80	73	73	**83**	80	77
	Modal - preterite	72	86	61	**88**	**88**	**88**	74	67	68	90	92	54	78	**93**	92	86	89	86	86	86	89	81	**94**	90	83
	Modal - preterite subjunctive II	30	**87**	80	83	83	83	77	80	80	**93**	87	43	73	87	**93**	83	80	**87**	83	**87**	83	70	**87**	83	81
	Modal negated - future I	43	95	93	81	93	**100**	88	93	93	86	**93**	86	91	65	86	58	81	**98**	77	79	79	74	79	70	84
	Modal negated - future I subjunctive II	73	96	92	86	90	**97**	96	90	90	79	92	79	88	77	**95**	84	86	**97**	79	82	88	75	86	67	87

Table 4: Accuracy of the metrics(%) with regards to the linguistically-motivated phenomena for German-English

(Continued on next page)

			baselines								QE as a metric							ref. based metrics								
ling. category	ling. phenomenon	#	BERTScore	BLEU	BLEURT-20	COMET-20	YiSi-1	chrF	f101spBLEU	f200spBLEU	COMETKiwi	Cross-QE	HWTSC-TLM	HWTSC-TS	KG-BERT	MS-COMET-QE	UniTE-src	COMET-22	MS-COMET	UniTE-ref	UniTE	XL-DA	XL-MQM	xxl-DA19	xxl-MQM20	avg
	Modal negated - perfect	126	71	50	66	**73**	**72**	62	52	52	73	**88**	60	71	63	85	72	63	**70**	63	66	63	60	63	51	66
	Modal negated - pluperfect	126	94	87	90	96	94	**99**	87	90	74	**95**	83	93	55	75	79	**88**	85	84	85	**88**	75	81	76	85
	Modal negated - pluperfect subjunctive II	81	75	65	73	72	**78**	74	69	68	59	**84**	64	79	**84**	**84**	73	73	**86**	74	72	79	75	74	69	74
	Modal negated - present	33	70	**79**	73	70	70	64	45	45	64	**88**	48	67	64	61	58	67	67	70	73	79	76	**82**	79	68
	Modal negated - preterite	61	90	66	90	**92**	89	87	56	56	90	82	38	75	**95**	**95**	90	85	**92**	85	87	80	79	80	84	81
	Modal negated - preterite subjunctive II	77	88	66	**91**	87	86	83	65	65	91	**95**	47	75	86	88	84	84	**91**	87	87	83	78	83	84	82
	Progressive	76	**84**	66	71	75	75	67	67	67	75	67	50	64	64	70	**76**	75	76	75	78	67	68	**79**	76	71
	Reflexive - future I	85	81	76	**89**	87	82	80	74	74	86	85	84	81	75	78	**88**	89	89	**92**	88	87	81	88	87	84
	Reflexive - future I subjunctive II	96	82	70	79	77	**84**	66	66	65	78	**89**	71	79	80	74	85	**85**	73	**86**	**85**	**85**	80	84	82	79
	Reflexive - future II	116	**97**	83	77	81	**97**	85	81	80	67	73	40	43	72	**75**	67	**87**	69	83	84	79	74	82	79	76
	Reflexive - future II subjunctive II	107	**93**	74	81	89	**93**	77	71	70	79	**92**	66	77	**91**	82	87	**89**	76	87	86	85	78	79	75	82
	Reflexive - perfect	188	81	64	81	**84**	**82**	62	69	68	86	85	53	54	78	72	**88**	86	80	**87**	85	82	78	82	83	77
	Reflexive - pluperfect	109	85	63	83	**88**	**87**	55	63	62	80	**83**	54	47	75	78	82	85	86	85	85	83	81	88	**92**	77
	Reflexive - pluperfect subjunctive II	90	**98**	76	79	87	**97**	80	78	78	70	81	66	70	**88**	76	81	**81**	76	**80**	**80**	74	64	77	71	79
	Reflexive - present	125	81	59	**90**	86	80	74	70	70	88	92	72	75	74	**94**	**94**	86	**92**	88	87	85	85	89	85	82
	Reflexive - preterite	117	86	69	85	83	**88**	75	70	71	76	83	54	56	66	**85**	83	88	76	**90**	**90**	**91**	85	85	83	79
	Reflexive - preterite subjunctive II	124	**92**	77	86	85	**91**	70	75	75	72	**83**	54	55	65	79	81	**89**	78	**89**	**88**	**89**	**88**	**88**	87	80
	Transitive - future I	43	98	95	**100**	**100**	**100**	95	95	95	**100**	95	86	**100**	**100**	95	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	98
	Transitive - future I subjunctive II	37	**100**	81	95	**100**	**100**	84	86	86	92	86	54	89	**100**	95	97	95	95	95	**97**	95	84	**97**	**97**	91
	Transitive - future II	33	**100**	76	94	94	**100**	94	79	79	88	64	70	**94**	88	**94**	76	94	88	94	94	**97**	85	**97**	85	88
	Transitive - future II subjunctive II	50	**100**	84	88	94	**100**	88	82	80	92	90	90	**98**	**98**	**98**	94	**92**	**92**	90	90	**92**	76	90	84	91
	Transitive - perfect	99	85	64	81	**88**	**88**	80	67	74	79	76	73	86	78	71	**93**	81	**90**	80	80	75	81	87	86	80
	Transitive - pluperfect	22	**91**	73	82	**91**	**91**	82	73	73	73	77	73	77	68	77	**91**	**91**	86	86	82	86	73	77	64	80
	Transitive - pluperfect subjunctive II	39	**100**	85	64	85	**100**	97	87	87	49	54	69	67	**92**	87	54	72	**92**	74	74	74	62	72	67	77
	Transitive - present	33	**94**	58	**94**	85	91	73	58	61	88	**94**	82	79	88	**94**	88	91	91	91	88	**94**	91	**94**	91	85
	Transitive - preterite	57	82	51	**86**	**86**	82	63	68	67	**95**	91	67	68	93	93	**95**	**100**	89	89	86	91	**100**	95	**100**	84
	Transitive - preterite subjunctive II	97	82	40	**86**	80	84	60	57	54	73	80	73	74	**86**	79	**85**	**85**	**86**	84	84	84	84	84	**85**	77
Verb valency	Case government	80	**89**	65	**88**	86	**89**	62	64	64	**94**	75	71	66	52	82	85	**95**	86	92	91	92	92	92	92	81
	Mediopassive voice	50	82	64	82	**84**	80	66	62	60	**74**	66	50	50	64	60	68	**90**	82	88	86	88	88	86	82	74
	Passive voice	33	**94**	85	91	**94**	**94**	82	82	79	**94**	79	64	61	64	82	91	**94**	**94**	**94**	**94**	91	91	88	91	86
	Resultative predicates	48	**100**	73	94	90	98	85	77	79	81	67	69	75	73	73	**83**	**96**	92	92	94	94	79	94	85	84
macro avg.		10402	**86**	71	83	83	**86**	76	72	72	79	**82**	65	71	73	77	79	**84**	82	**84**	**84**	83	79	**84**	80	79
micro avg.		10402	**84**	70	82	82	**85**	75	72	72	78	**81**	66	70	70	75	78	**84**	80	**83**	82	82	79	82	79	78

Table 4: Accuracy of the metrics(%) with regards to the linguistically-motivated phenomena for German-English

ling. category	ling. phenomenon	#	baselines								QE as a metric									ref. based metrics												
			BERTScore	BLEU	BLEURT-20	COMET-20	YiSi-1	chrF	f101spBLEU	f200spBLEU	COMETKiwi	Cross-QE	HWTSC-TLM	HWTSC-TS	KG-BERT	MATESE-QE	MS-COMET-QE	REUSE	UniTE-src	COMET-22	MATESE	MEE	MEE2	MEE4	MS-COMET	UniTE-ref	UniTE	XL-DA	XL-MQM20	XXL-DA19	XXL-MQM20	avg
Ambiguity	Lexical ambiguity	146	87	71	**90**	82	87	**89**	87	88	55	47	**81**	47	47	25	38	15	36	84	40	73	88	91	78	**97**	93	94	88	95	87	72
Coordination	Gapping	163	72	58	**84**	**84**	74	68	67	64	**80**	71	71	72	42	61	78	58	**81**	79	58	62	66	64	78	81	77	79	88	**91**	86	72
& ellipsis	Pseudogapping	201	82	77	**97**	87	82	67	76	74	93	92	81	78	78	41	75	26	**94**	96	53	65	73	74	89	**97**	**97**	**97**	93	95	93	80
	Right node raising	47	83	64	87	**91**	83	72	72	70	**87**	81	81	83	34	4	**87**	9	79	94	4	62	66	66	91	89	94	91	**96**	94	89	73
	Sluicing	169	54	56	**63**	57	59	54	59	58	56	47	61	51	46	14	**72**	51	53	62	17	63	59	56	59	57	55	**67**	**67**	56	65	55
	Stripping	139	66	58	65	60	**68**	58	58	60	**63**	53	53	55	55	22	41	38	55	68	29	45	58	58	71	71	68	67	**83**	73	73	58
	VP-ellipsis	117	59	47	85	**87**	75	51	46	46	84	85	**95**	91	92	29	78	26	82	82	30	54	50	48	**91**	85	84	85	80	80	76	69
False friends	False friends	225	66	63	70	**73**	67	72	66	67	67	60	64	73	68	52	73	**89**	64	69	35	69	79	**88**	76	71	71	71	71	68	69	69
Function	Focus particle	20	45	30	80	**90**	45	35	30	30	45	50	15	25	25	5	**70**	55	65	75	5	30	30	35	50	70	70	60	55	**85**	75	48
word	Question tag	180	**94**	82	91	**95**	82	76	79	78	96	**97**	84	**98**	**98**	72	94	68	**97**	92	31	83	86	91	**94**	92	**93**	80	85	83	83	85
MWE	Collocation	112	73	61	**92**	79	88	76	62	61	**88**	86	**89**	86	86	46	75	46	86	**95**	50	68	79	80	85	**95**	93	93	**96**	93	**96**	79
	Compound	63	75	51	70	78	**87**	84	71	71	98	**100**	89	94	94	21	57	3	86	**97**	22	57	70	67	73	**97**	90	90	**97**	81	84	74
	Idiom	266	86	82	**95**	**95**	92	75	80	81	86	**92**	**93**	85	86	62	75	22	82	**98**	67	77	82	84	96	**98**	**98**	97	97	97	95	85
	Nominal MWE	288	81	71	**84**	72	81	78	78	74	62	72	66	**78**	**78**	6	71	39	74	69	28	65	72	73	68	74	**76**	72	74	55	48	67
	Prepositional MWE	35	69	**86**	71	83	**86**	83	74	80	66	77	60	80	80	60	**89**	83	80	80	66	77	80	83	**89**	80	80	86	86	80	83	78
	Verbal MWE	65	66	71	**89**	78	62	74	58	58	**83**	58	69	65	65	57	55	23	58	86	42	62	74	75	72	78	80	85	75	**92**	88	69
Named entity	Date	234	55	53	**74**	66	68	60	61	65	62	**93**	80	79	79	48	80	31	50	65	54	62	60	57	64	72	**75**	67	**76**	70	73	65
& termin.	Domainspecific term	312	73	56	**89**	**90**	86	76	69	71	73	62	**78**	67	67	7	58	33	71	78	41	67	73	76	93	86	85	**97**	94	**96**	92	73
	Location	12	67	83	75	**100**	50	58	58	83	**100**	92	**100**	92	**100**	0	92	83	92	75	17	92	83	92	83	83	92	**100**	**100**	**100**	**100**	81
	Measuring unit	389	54	48	53	50	54	**55**	**57**	53	28	31	21	20	20	6	37	**43**	35	46	12	51	57	59	41	57	57	**69**	58	**69**	68	45
	Proper name	325	50	61	52	51	53	54	66	**69**	**64**	**64**	58	58	61	34	53	62	58	59	23	58	61	61	56	**64**	59	58	54	58	55	56
Negation	Negation	174	87	83	89	90	**93**	85	82	84	**92**	86	87	**91**	**91**	43	**92**	78	90	91	79	84	92	92	90	**94**	**94**	82	81	82	78	86
Non-verbal	Coreference	81	85	86	**95**	84	75	86	84	89	**77**	73	33	67	51	26	**77**	41	73	96	80	81	89	88	95	94	93	96	**100**	**99**	**99**	80
agreement	Genitive	206	76	73	73	**83**	82	68	77	76	71	63	83	56	56	44	**90**	22	62	84	42	57	70	72	81	82	82	85	85	**87**	**87**	71
	Possession	85	61	55	**86**	**85**	74	58	58	60	**93**	86	26	76	76	19	53	78	**93**	**96**	33	51	62	72	79	93	92	88	91	87	89	71
Punctuation	Quotation marks	336	70	**79**	76	77	77	74	71	68	68	72	70	51	51	50	68	46	**79**	79	51	64	75	74	73	**81**	**81**	67	60	72	68	69
Subordination	Adverbial clause	193	72	**81**	**81**	**81**	67	73	79	77	88	77	79	87	87	34	72	65	**90**	82	31	72	78	77	71	84	82	**86**	**85**	82	**85**	76
	Cleft sentence	179	66	63	60	**69**	63	57	62	60	74	59	72	**82**	**82**	45	74	45	67	71	46	59	65	68	**73**	70	**73**	**73**	66	65	63	65
	Contact clause	150	83	75	**94**	**94**	88	74	74	73	**98**	**97**	**99**	97	97	65	92	53	96	**98**	64	79	76	79	92	**97**	96	**97**	**97**	95	93	87
	Indirect speech	38	58	42	63	**66**	50	47	47	42	**95**	63	58	58	50	42	55	24	55	**76**	47	39	42	42	74	71	71	63	68	63	61	56
	Infinitive clause	85	67	55	86	87	**95**	80	66	66	95	**99**	78	79	68	66	**98**	40	**98**	93	65	54	67	71	**95**	93	92	88	91	**94**	87	80
	Object clause	16	75	38	**88**	**88**	62	56	38	38	81	62	56	81	81	62	**88**	31	81	88	50	44	50	50	94	**100**	**100**	**100**	**100**	88	75	70
	Pseudo-cleft sentence	73	**90**	88	66	70	**90**	**89**	82	82	68	85	81	**89**	**89**	62	73	70	78	75	60	**93**	**92**	**93**	58	71	77	85	86	68	68	79
	Relative clause	112	89	83	90	**94**	82	84	88	81	88	93	78	65	65	22	96	36	**100**	91	52	85	87	86	96	**98**	96	94	89	84	79	82

Table 5: Accuracy of the metrics(%) with regards to the linguistically-motivated phenomena for German-English

(Continued on next page)

			baselines								QE as a metric									ref. based metrics												
ling. category	ling. phenomenon	#	BERTScore	BLEU	BLEURT-20	COMET-20	YiSi-1	chrF	f101spBLEU	f200spBLEU	COMETKiwi	Cross-QE	HWTSC-TLM	HWTSC-TS	KG-BERT	MATESE-QE	MS-COMET-QE	REUSE	UniTE-src	COMET-22	MATESE	MEE	MEE2	MEE4	MS-COMET	UniTE-ref	UniTE	XL-DA	XL-MQM	xxl-DA19	xxl-MQM20	avg
Verb tense /aspect/mood	Subject clause	148	86	**90**	89	**91**	**91**	**90**	**91**	**91**	**89**	89	87	**89**	**89**	47	**91**	33	85	86	71	87	**92**	**93**	89	89	88	84	82	89	85	85
	Conditional	106	74	77	**94**	90	91	70	75	75	**92**	87	86	89	89	31	81	18	**92**	87	52	75	83	87	88	**92**	**92**	**92**	84	89	89	80
	Ditransitive - conditional I progr.	72	65	49	**93**	89	83	61	56	57	**99**	**99**	94	**99**	79	50	92	74	**100**	92	47	64	60	62	81	90	92	93	81	**100**	89	79
	" - conditional I simple	34	94	74	65	85	**97**	94	74	79	**100**	97	41	41	44	26	91	91	**100**	**100**	18	**100**	94	97	85	**100**	97	97	94	**100**	91	82
	" - conditional II progr.	51	75	78	**88**	78	80	82	82	82	65	67	51	55	49	24	59	63	**86**	90	27	86	82	78	82	88	86	84	82	86	**92**	73
	" - conditional II simple	59	71	64	76	**78**	66	68	64	73	69	56	53	49	47	36	63	59	**78**	73	25	78	71	75	78	78	76	80	78	**81**	78	67
	" - future I progr.	61	52	51	**62**	57	57	**62**	51	51	92	51	84	75	49	11	80	90	**97**	66	8	59	59	61	33	59	66	**79**	75	57	66	61
	" - future I simple	88	**60**	51	56	55	56	**60**	50	45	66	50	52	53	48	40	70	**90**	85	58	38	56	57	60	53	56	60	**65**	**64**	51	60	57
	" - future II progr.	91	**70**	64	66	57	47	60	65	62	71	45	84	**91**	**91**	11	78	56	77	82	14	89	73	76	54	86	77	65	62	**95**	92	68
	" - future II simple	49	71	**94**	86	**94**	65	**94**	92	92	**100**	92	76	71	65	8	65	88	92	92	18	96	94	94	65	**100**	98	86	39	88	76	79
	" - past perfect progr.	91	60	44	60	**67**	**66**	58	53	48	65	**75**	51	59	65	11	**75**	37	60	71	33	62	59	63	52	67	75	**78**	73	73	67	60
	" - past perfect simple	112	63	62	65	56	**72**	71	61	61	56	**79**	37	37	54	8	58	43	46	70	39	56	64	64	53	62	68	**71**	57	58	49	57
	" - past progr.	83	58	57	**70**	58	59	61	57	57	61	37	39	37	37	12	42	**71**	33	70	12	39	55	60	42	66	64	**72**	67	**72**	69	53
	" - present perfect progr.	48	85	54	85	75	**92**	88	56	60	85	94	90	**100**	**100**	21	77	52	**100**	92	35	75	81	81	71	**94**	**94**	79	73	92	77	78
	" - present perfect simple	54	**65**	37	56	43	30	41	37	44	33	33	31	26	**35**	28	33	33	31	48	22	44	44	48	33	57	56	65	59	**70**	69	43
	" - present progr.	72	76	38	94	**97**	90	68	36	49	**100**	**100**	**99**	**99**	**99**	94	88	35	**99**	**96**	71	72	86	83	**97**	**97**	**97**	88	88	92	88	83
	" - simple past	77	77	56	77	**83**	56	66	56	57	**97**	94	69	75	88	36	73	82	82	**94**	45	73	78	84	87	82	83	79	84	82	78	75
	" - simple present	54	72	30	**83**	70	**83**	56	41	41	67	**70**	67	67	67	54	**70**	28	**70**	59	48	56	67	69	65	80	80	81	83	89	**94**	66
	Gerund	161	92	85	**96**	**96**	92	80	83	82	97	**99**	58	87	87	19	97	78	**99**	**97**	25	83	85	88	**98**	96	96	96	96	**97**	87	85
	Imperative	50	70	50	**96**	94	70	70	58	64	**100**	92	78	86	86	80	94	82	88	**96**	60	70	70	76	94	92	92	**96**	90	94	92	82
	Intransitive - conditional I progr.	9	56	89	89	**100**	**100**	78	78	89	**100**	44	0	22	22	67	44	**100**	**100**	89	56	78	78	89	78	**100**	**100**	33	56	89	78	72
	" - conditional I simple	3	**100**	0	67	**100**	**100**	33	0	33	**100**	**100**	33	33	33	**100**	33	**100**	67	**100**	**100**	0	67	**100**	67	67	67	67	**100**	**100**	67	67
	" - future I progr.	7	71	86	**100**	**100**	57	**100**	86	86	57	57	0	29	29	**86**	57	71	71	86	0	71	86	**100**	57	**100**	**100**	71	**100**	**100**	**100**	73
	" - future I simple	24	67	**75**	**75**	71	50	67	67	71	96	**100**	71	46	46	29	92	96	**100**	62	42	58	**67**	**67**	**67**	58	62	58	58	**67**	**67**	67
	" - future II progr.	4	25	**50**	25	25	**50**	**50**	**50**	**50**	**75**	0	**75**	25	25	0	50	25	0	50	0	75	50	25	25	25	25	50	75	50	**100**	40
	" - future II simple	7	71	**100**	86	**100**	**100**	**100**	**100**	**100**	**100**	**100**	57	71	71	0	43	**100**	71	**100**	14	71	86	86	**100**	86	86	43	43	71	57	76
	" - past perfect progr.	16	56	50	38	62	69	62	**81**	69	50	69	38	44	44	0	**75**	38	44	50	6	56	62	62	**69**	31	31	56	38	62	44	50
	" - past perfect simple	18	78	72	**89**	72	61	78	61	61	**94**	50	89	78	78	0	56	44	39	83	17	**89**	78	78	78	83	72	78	67	**89**	78	69
	" - past progr.	28	43	57	**71**	**71**	54	57	54	54	**68**	50	46	36	36	29	61	46	57	50	25	50	**61**	54	50	57	57	57	54	**61**	57	52
	" - present perfect simple	2	**100**	50	**100**	**100**	**100**	**100**	50	50	**100**	**100**	**100**	**100**	**100**	0	**100**	**100**	**100**	**100**	0	50	**100**	**100**	**100**	**100**	**100**	**100**	50	**100**	**100**	84
	" - present progr.	5	80	**100**	80	80	80	**100**	80	80	**80**	**80**	0	0	0	20	**80**	60	**80**	80	60	80	**100**	**100**	80	80	80	**100**	80	80	80	72
	" - simple past	24	58	38	**62**	58	58	46	38	38	**100**	**100**	96	**100**	**100**	46	71	96	88	71	46	38	62	71	67	83	**88**	62	58	79	79	69
	" - simple present	10	40	30	**50**	40	40	40	40	40	**70**	**70**	40	60	60	**70**	**70**	40	**70**	**70**	50	20	30	30	**70**	50	50	60	50	50	40	50

Table 5: Accuracy of the metrics(%) with regards to the linguistically-motivated phenomena for German-English

(Continued on next page)

ling. category	ling. phenomenon	#	baselines								QE as a metric									ref. based metrics												
			BERTScore	BLEU	BLEURT-20	COMET-20	YiSi-1	chrF	f101spBLEU	f200spBLEU	COMETKiwi	Cross-QE	HWTSC-TLM	HWTSC-TS	KG-BERT	MATESE-QE	MS-COMET-QE	REUSE	UniTE-src	COMET-22	MATESE	MEE	MEE2	MEE4	MS-COMET	UniTE-ref	UniTE	XL-DA	XL-MQM	xxl-DA19	xxl-MQM20	avg
	Modal	20	**70**	60	40	60	45	55	60	55	10	15	50	60	60	0	**100**	**100**	90	25	0	55	70	**75**	65	40	40	35	35	20	20	49
	Modal negated	20	35	65	70	**75**	65	60	65	70	65	**95**	50	**95**	**95**	0	65	70	60	85	0	55	70	70	60	80	80	**95**	80	90	85	67
	Reflexive - conditional I progr.	65	66	52	48	45	45	46	**71**	**71**	38	63	15	23	23	**71**	46	28	63	52	58	74	63	54	37	51	54	83	**85**	60	58	53
	" - conditional I simple	112	**76**	70	48	67	58	70	72	72	32	**100**	9	27	27	76	43	37	60	64	80	**90**	80	66	48	57	62	86	**89**	78	72	63
	" - conditional II progr.	97	**71**	**72**	66	67	61	69	**71**	**71**	64	**80**	10	20	20	76	24	49	55	84	77	86	84	67	61	72	73	87	**89**	78	86	65
	" - conditional II simple	109	61	**68**	52	55	54	61	58	59	50	**92**	11	21	27	81	16	28	57	78	86	78	67	47	59	53	49	83	91	84	**93**	59
	" - future I progr.	70	67	67	70	54	**84**	79	66	66	59	66	60	**77**	**77**	47	69	64	63	79	40	77	74	66	56	60	67	**80**	76	70	66	67
	" - future I simple	83	69	67	71	54	76	**86**	77	77	61	61	49	61	61	45	**78**	63	66	76	33	**78**	**78**	75	45	66	71	**78**	72	65	54	66
	" - future II progr.	81	65	56	64	73	75	**80**	57	57	73	**88**	54	63	63	81	51	53	65	83	62	73	79	72	58	70	73	**85**	80	68	60	68
	" - future II simple	56	71	66	77	61	**88**	**88**	64	64	79	**98**	61	59	59	55	39	68	75	88	54	**89**	80	75	55	62	68	79	71	79	70	70
	" - past perfect progr.	98	60	50	67	63	**71**	66	60	51	66	**82**	33	46	46	44	52	51	79	**76**	42	71	67	67	55	63	62	71	73	71	66	61
	" - past perfect simple	53	62	47	68	62	**74**	55	57	57	64	**98**	25	34	34	66	17	43	57	**87**	66	81	68	62	58	51	62	79	85	85	66	61
	" - past progr.	5	**100**	**100**	40	**100**	**100**	**100**	**100**	**100**	80	60	20	20	20	40	**100**	40	40	80	20	**100**	**100**	**100**	**100**	80	**100**	80	80	80	80	74
	" - present perfect progr.	33	76	48	**88**	82	76	76	48	48	**100**	**100**	64	61	61	**100**	45	24	82	**100**	**100**	97	82	79	58	79	82	97	**100**	91	85	77
	" - present perfect simple	39	59	46	67	69	69	**72**	44	44	74	**92**	79	72	72	21	31	54	69	85	74	77	69	69	51	54	69	**87**	85	74	77	66
	" - present progr.	99	62	51	54	54	**67**	56	60	62	36	**77**	27	26	26	61	40	46	45	58	48	68	61	57	40	53	56	62	**70**	54	63	53
	" - simple past	119	71	70	73	76	73	**77**	71	71	**89**	83	37	69	76	40	46	53	76	**91**	39	81	75	71	73	74	76	82	76	83	81	71
	" - simple present	138	65	65	67	62	**88**	63	68	67	44	**89**	39	54	62	62	47	32	49	62	46	**78**	76	69	69	54	57	71	69	68	67	62
	Transitive - future II progr.	11	73	**82**	73	73	64	**82**	**82**	**82**	73	55	82	**91**	**91**	9	**91**	73	**91**	82	9	73	73	73	82	73	82	91	82	**100**	**100**	75
	" - conditional I progr.	11	55	**91**	45	73	36	82	**91**	**91**	55	18	36	45	45	0	**82**	**82**	27	45	0	55	55	55	**73**	45	45	45	27	27	18	50
	" - conditional I simple	9	67	**100**	89	89	56	**100**	**100**	**100**	**100**	67	56	67	67	0	89	67	67	**100**	33	67	67	67	**100**	78	56	78	44	67	67	72
	" - conditional II progr.	20	70	55	75	70	**80**	55	60	60	**75**	40	35	50	50	0	40	60	35	85	0	55	65	75	75	60	60	70	65	**100**	**100**	59
	" - conditional II simple	2	50	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	50	50	50	50	0	**100**	50	**100**	**100**	0	**100**	**100**	**100**	**100**	**100**	**100**	**100**	50	**100**	**100**	81
	" - future I progr.	12	42	**83**	75	67	25	50	75	75	**75**	50	50	42	42	42	58	50	42	67	25	50	50	50	**83**	42	42	42	17	50	58	52
	" - future I simple	22	64	**95**	64	64	59	77	91	**95**	36	50	41	18	18	18	**82**	50	50	55	23	68	68	68	68	45	45	59	36	64	**82**	57
	" - future II simple	39	62	**92**	59	72	67	85	90	90	**82**	**82**	64	72	72	3	69	46	79	69	10	77	**82**	**82**	69	74	67	72	38	67	54	67
	" - past perfect progr.	16	50	69	50	56	**81**	**81**	69	69	62	**75**	38	38	38	6	**75**	56	62	**75**	25	44	62	56	69	31	44	62	38	62	38	55
	" - past perfect simple	9	44	78	**89**	78	33	**89**	78	78	**100**	56	89	78	78	0	56	44	**100**	**89**	67	33	44	44	**89**	67	44	78	78	78	44	66
	" - present perfect progr.	5	20	**80**	**80**	**80**	20	**80**	**80**	**80**	40	40	**100**	60	60	0	**100**	20	20	60	60	20	20	20	**100**	60	60	60	20	60	20	52
	" - present perfect simple	9	33	67	**78**	56	44	**78**	67	67	78	33	**100**	78	78	0	**100**	44	89	**78**	44	22	33	33	**78**	67	44	67	22	67	44	58
	" - present progr.	10	30	**70**	20	30	30	40	50	50	**50**	40	40	40	40	20	40	0	20	30	20	40	40	40	40	40	40	**50**	**50**	30	40	37
	" - simple past	23	61	43	**96**	78	35	57	48	52	87	52	61	57	57	13	**91**	61	78	87	52	30	57	65	78	87	78	**91**	70	83	**91**	65
	" - simple present	16	31	62	38	44	**69**	62	56	56	94	44	31	31	31	44	**100**	50	62	**81**	31	31	38	38	69	50	44	50	25	62	62	51
Verb valency	Case government	57	**82**	67	75	79	**82**	70	70	75	**86**	79	72	77	68	75	72	44	74	**82**	65	63	77	77	63	81	**82**	77	75	81	79	74

Table 5: Accuracy of the metrics(%) with regards to the linguistically-motivated phenomena for German-English

(Continued on next page)

			baselines								QE as a metric									ref. based metrics												
ling. category	ling. phenomenon	#	BERTScore	BLEU	BLEURT-20	COMET-20	YiSi-1	chrF	f101spBLEU	f200spBLEU	COMETKiwi	Cross-QE	HWTSC-TLM	HWTSC-TS	KG-BERT	MATESE-QE	MS-COMET-QE	REUSE	UniTE-src	COMET-22	MATESE	MEE	MEE2	MEE4	MS-COMET	UniTE-ref	UniTE	XL-DA	XL-MQM	xxl-DA19	xxl-MQM20	avg
	Catenative verb	177	69	58	**86**	61	70	62	60	60	**77**	67	71	71	71	25	60	28	60	76	29	62	64	62	65	68	70	67	72	**89**	**89**	64
	Middle voice	29	90	69	**93**	**93**	79	83	83	83	79	76	**90**	83	83	21	48	31	62	83	45	83	90	**97**	**97**	**97**	**97**	**97**	93	93	86	79
	Passive voice	70	64	51	67	**74**	66	71	53	61	**87**	74	76	71	71	21	70	47	70	**87**	43	50	61	63	86	71	70	79	71	76	77	67
	Resultative	147	76	74	**90**	85	86	80	73	80	84	80	45	61	59	24	84	76	**88**	88	48	63	73	76	87	**92**	**91**	89	87	84	73	76
macro avg.		8945	67	65	**74**	**74**	70	70	66	67	**75**	72	60	62	61	35	69	53	71	**79**	39	65	70	70	72	74	75	**77**	73	**78**	74	68
micro avg.		8945	70	65	**76**	74	73	69	68	68	**73**	**74**	63	65	64	38	67	48	71	78	42	68	71	72	72	77	77	**79**	77	**78**	76	69

Table 5: Accuracy of the metrics(%) with regards to the linguistically-motivated phenomena for English-German

Exploring Robustness of Machine Translation Metrics: A Study of Twenty-Eight Automatic Metrics in the WMT22 Metric Task

Xiaoyu Chen, Daimeng Wei, Hengchao Shang, Zongyao Li, Zhanglin Wu,
Zhengzhe Yu, Ting Zhu, Mengli Zhu, Ning Xie, Lizhi Lei, Shimin Tao,
Hao Yang, Ying Qin
Huawei Translation Services Center, Beijing, China
{chenxiaoyu35,weidaimeng,shanghengchao,lizongyao,wuzhanglin2,
yuzhengzhe,zhuting20,zhumengli,nicolas.xie,leilizhi,taoshimin,
yanghao30,qinying}@huawei.com

Abstract

Contextual word embeddings extracted from pre-trained models have become the basis for many downstream NLP tasks, including machine translation automatic evaluations. Metrics that leverage embeddings claim better capture of synonyms and changes in word orders, and thus better correlation with human ratings than surface-form matching metrics (e.g. BLEU). However, few studies have been done to examine robustness of these metrics. This report uses a challenge set to uncover the brittleness of reference-based and reference-free metrics. Our challenge set[1] aims at examining metrics' capability to correlate synonyms in different areas and to discern catastrophic errors at both word- and sentence-levels. The results show that although embedding-based metrics perform relatively well on discerning sentence-level negation/affirmation errors, their performances on relating synonyms are poor. In addition, we find that some metrics are susceptible to text styles so their generalizability compromised.

1 Introduction

Automatic metrics compare machine-translated results with human-translated references or/and sources, and give scores accordingly. Such metrics offer a quick and inexpensive approach for researchers to evaluate model performances. Among these metrics, BLEU (Papineni et al., 2002) has dominated the area for twenty years since its birth in 2002. However, its limitations are obvious: (1) it weighs each word equally but in fact the entropy of each word varies; (2) it only counts n-grams that are exact in the reference and thus synonyms and elaborations are wrongly punished (Smith et al., 2016). Consequently, the correlation between BLEU and human evaluation is relatively low, which sometimes puzzles researchers.

In recent years, embedding-based approaches have been introduced to design new automatic metrics. These metrics, e.g. BERTScore (Zhang et al., 2019), COMET (Rei et al., 2020a), and BLEURT (Sellam et al., 2020a), claim better ability to capture synonyms and changes in word order, and thus better performance than BLEU. Apart from ref-based metrics, researches on quality estimation (QE) have been rising, as QE is an cheaper and more convenient approach considering no need of human-translated references.

In the WMT metric task, correlation with human annotators is the major indicator to evaluate metric performance (Freitag et al., 2021). However, in addition to that, a good metric should meet the following requirements (Banerjee and Lavie, 2005; Koehn, 2009): (1) sensitivity to nuances in quality among systems or outputs of the same system in different stages of its development so it can be used to direct system performance optimization; (2) consistency and reliability of scores; (3) usability in a great range of fields; (4) speed; (5) low cost. We believe the first three aforementioned requirements are crucial for judging metric performance as well. So we build a Zh→En challenge set to evaluate metrics' capability in these regards. Section 2 offers a brief description of metrics to be evaluated. Details of our challenge set are described in Section 3. Section 4 presents experiment results and Section 5 discusses our findings.

2 Metrics To Be Evaluated

2.1 Surface-Form Matching Metrics

Reference-based metrics measure the similarity between MT outputs and human translations, and believe that high similarity means high quality and vice versa. In the pre-neural era, metrics calculate the similarity based on surface forms and word stems. Two examples that fall into this category and used in this task as baselines are BLEU (Papineni

[1] We open-source our challenge set at: https://github.com/HwTsc/Challenge-Set-for-MT-Metrics

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 530 - 540
December 7-8, 2022 ©2022 Association for Computational Linguistics

et al., 2002) and chrF (Popović, 2015).

BLEU BLEU computes precision by comparing the n-gram of hypothesis with n-gram of the reference, coupled with a brevity penalty. In this task, sentence-level BLEU (SENT-BLEU) is used.

chrF chrF computes F1 score based on character-level n-grams instead of word-level n-grams.

2.2 Embedding-based Metrics

In the neural era, by leveraging pre-trained word embeddings, new metrics claim better understanding of sentence meanings and thus fare better in evaluation tasks. Some of the well-known metrics that fall into this category and used as baselines in this task include:

BERTScore BERTScore (Zhang et al., 2019) outputs F1 score by calculating token similarity based on contextual embeddings extracted from BERT.

BLEURT-20 BLEURT (Sellam et al., 2020a) is a BERT-based regression model trained on rating data. BLEURT-20 (Sellam et al., 2020b), which is fine-tuned based on Rebalanced mBERT is used in this task.

COMET-20 COMET (Rei et al., 2020a) employs the estimator-predictor architecture and leverages both source and reference information to assess translation quality. COMET-20 (Rei et al., 2020b), which utilizes XLM-RoBERTa, is used in this task.

Yisi-1 Yisi (Lo, 2019) measures semantic similarity between hypothesis and references. Yisi-1 (Lo, 2020) leverages contextual embeddings extracted from language models to compute the idf-weighted lexical semantic similarities.

2.3 QE as Metrics

Quality estimation approach evaluates machine translation quality totally without human intervention. It scores model outputs by leveraging information in source text. Among the seven baseline metrics, COMET-QE (Rei et al., 2021) is a reference-free version of COMET and thus falls into the QE category.

Table 1 is a summary of the seven baselines.

Metrics	Surface	CWE	Source	Ref	Rating
BLEU	Yes	No	No	Yes	No
chrF	Yes	No	No	Yes	No
BERTScore	No	Yes	No	Yes	No
BLEURT-20	No	Yes	No	Yes	Yes
COMET-20	No	Yes	Yes	Yes	Yes
COMET-QE	No	Yes	Yes	No	Yes
YISI-1	No	Yes	No	Yes	No

Table 1: A comparison of seven baseline metrics from aspects of whether they use surface form (Surface), contextual word embedding (CWE), Source text (Source), Target text (Ref), and human rating data (Rating).

2.4 Participants in WMT22 Metric Task

The challenge set is also used to measure performances of metrics submitted to the WMT22 Metric Task, including twelve reference-based (ref-based) metrics: COMET-22, MATASE, three variants of MEE, four variants of Metricx, ME-COMET-22, two variants of UniTE; and nine QE metrics: COMET-Kiwi, Cross-QE, HWTSC-Teacher-Sim, HWTSC-TLM, KG-BERTScore, MATESE-QE, MS-COMET-QE, REUSE, and UniTE-src.

For details about their implementations, please refer to their system reports and summary report of WMT22 Metrics Task[2].

3 Challenge Set & Method

3.1 Source of the Challenge Set

We build our Zh-En challenge set to evaluate metrics' ability to relate synonyms and identify crucial mistakes. The set is built based on two open-source test sets: Flores 101 (Goyal et al., 2022) En-Zh subset (but used as a Zh-En test set in this task) and WMT21 Zh-En news dev + test sets (Akhbardeh et al., 2021). We particularly pick up an En-Zh test set and a Zh-En test set because neural-based metrics may be style-sensitive (Hanna and Bojar, 2021): the English side of the En-Zh test set is natural language while that of the Zh-En test set is translation results, which may suffer from translationese. In addition, WMT sets focus on news domain while Flores is extracted from Wiki. We try to understand whether reference style and domain might influence metric performance so as to evaluate the generalizability of metrics.

3.2 Challenge Set Description

Our test set has 721 test cases and focuses on five categories of errors: (1) number; (2) date & time (D/T); (3) named-entity & terminology (NE&Term); (4) unit; and (5) affirmation/negation

[2] At the time of writing, we have not received descriptions from every participant.

Phenomenon	Flores	WMT	Overall
Number	183	172	355
D/T	50	90	140
NE&Term	68	42	110
Unit	23	35	58
AFF/NEG	58	0	58
Overall	382	339	721

Table 2: Challenge set composition

(AFF/NEG). Each case contains a source text, a reference, a good translation, a bad translation, a language phenomena label and a source of origin label indicating where the sentence comes from. Table 2 details the set composition. The first four categories focus on word-level crucial errors. If such information is translated wrong, human annotators will assign relatively low scores since the audience will be misled by such mistakes. In addition, the four categories feature rich types of expressions. For instance, a number can be presented in either numeral or number format; unit, named entity and terminology have widely-used abbreviations. We try to analyze whether metrics are able to relate synonyms and punish errors the way human annotators do. The last category – affirmation/negation – deals with phrase- to sentence-level errors and tests whether metrics are able to capture the overall meaning of a sentence.

Since both sets provide only one translation result for each sentence, to generate an additional translation result, we employ a group of six in-house translators to post-edit MT results generated by our in-house model. We adopt List-based Attack (LIST) (Alzantot et al., 2018) to generate adversarial examples. LIST replaces word(s) in a candidate sentence with a list of similar words to construct adversarial examples. We use semi-auto and human-craft approaches to extract related sentences from the original data sets. We replace key words in those sentences to ensure that key information in references and good examples are semantically equal but in different formats, and that in references and adversarial examples are semantically different but in the same "surface" format.

Table 3 shows an example of our challenge set. The test case contains a source sentence, a reference, a good-translation that contains a correct translation for a language phenomenon, and an incorrect-translation with an error accordingly. Phenomenon to be evaluated and source of the sentence are also labelled in our challenge set. In this case, it is number and comes from Flores. For more examples, please see Appendix A.

SRC:	在已知的大约24,000块坠落至地球的陨石中，经核实只有***34***块是来自火星。
REF:	Out of the approximately 24,000 known meteorites to have fallen to Earth, only about ***34*** have been verified to be martian in origin.
GOOD:	Of the roughly 24,000 meteorites known to have fallen to Earth, only ***thirty-four*** have been confirmed to have come from Mars.
BAD:	Of the roughly 24,000 meteorites known to have fallen to Earth, only ***30*** have been confirmed to have come from Mars.

Table 3: A case of number in different formats. GOOD refers to good translation and BAD refers to the adversarial example.

3.3 Measurement

Kendall's tau-like correlation (Freitag et al., 2021) is used to evaluate metric performance. A good translation has higher quality than the corresponding bad one, so a good metric should assign a higher score to the good translation. If a metric does so, we label the metric "Concordant" on the case, and "Discordant" vice versa. The correlation is calculated based on the following formula:

$$\tau = \frac{Concordant - Discordant}{Concordant + Discordant}$$

4 Result

Table 4 presents the results of our challenge set. In general, 20 out of the 28 metrics struggle to discriminate between good and adversarial examples as they fail to achieve a medium correlation (above 0.4) with human annotators. The 8 metrics that manage to achieve medium-level correlation including: BLEURT-20 (baseline), four variants of Metricx (ref-based), HWTSC-Teacher-Sim (QE), KG-BERTScore (QE), and REUSE (QE).

4.1 Comparison across types of metrics

In general, embedding-based metrics perform much better than merely surface-form matching

Metric	Overall	Number	D/T	NE&Term	Unit	AFF/NEG
SENT-BLEU	-0.717	-0.735	-0.743	-0.691	-0.621	-0.690
chrF	-0.393	-0.301	-0.300	-0.745	-0.655	-0.241
BERTScore	-0.193	-0.149	-0.429	-0.291	-0.483	0.586
BLEURT-20	0.495	0.476	0.629	0.364	0.310	0.724
COMET-20	-0.132	-0.093	-0.400	-0.200	-0.414	0.690
COMET-QE	0.090	0.048	-0.343	0.400	0.069	0.828
Yisi-1	-0.140	-0.138	-0.271	-0.291	-0.379	0.690
Baseline Avg.	**-0.141**	**-0.128**	**-0.265**	**-0.208**	**-0.310**	**-0.369**
COMET-22	0.331	0.206	0.500	0.327	0.138	0.897
MATESE	-0.476	-0.673	-0.429	-0.109	-0.414	-0.138
MEE	-0.667	-0.662	-0.700	-0.564	-0.897	-0.586
MEE2	0.060	0.251	0.229	-0.600	-0.345	0.138
MEE4	0.171	0.307	0.443	-0.473	-0.138	0.207
metricx_xl_DA	0.778	0.746	0.900	0.727	0.586	0.966
metricx_xl_MQM	0.781	0.685	0.900	0.818	0.828	0.966
metricx_xxl_DA	0.822	0.820	0.800	0.800	0.828	0.931
metricx_xxl_MQM	0.870	0.865	0.829	0.873	0.897	0.966
MS-COMET-22	0.012	-0.054	-0.143	0.055	-0.103	0.828
UniTE	0.287	0.177	0.500	0.200	-0.069	0.966
UniTE-ref	0.343	0.234	0.529	0.327	0.000	0.931
Ref-based Avg.	**0.276**	**0.242**	**0.363**	**0.198**	**0.109**	**0.589**
COMET-Kiwi	0.337	0.177	0.243	0.582	0.483	0.931
Cross-QE	0.340	0.245	0.171	0.473	0.448	0.966
HWTSC-Teacher-Sim	0.445	0.504	0.314	0.309	0.345	0.759
HWTSC-TLM	0.393	0.425	0.271	0.364	0.310	0.621
KG-BERTScore	0.445	0.493	0.286	0.491	0.138	0.759
MATESE-QE	-0.675	-0.735	-0.771	-0.400	-0.690	-0.586
MS-COMET-QE	0.146	0.059	0.114	0.127	0.000	0.931
REUSE	0.528	0.577	0.657	0.291	0.241	0.655
UniTE-src	0.268	0.104	0.314	0.473	0.103	0.931
QE Avg.	**0.247**	**0.206**	**0.178**	**0.301**	**0.153**	**0.663**

Table 4: Kendall's tau-like correlation results of each metric on our challenge set. The horizontal lines delimit baseline metrics (top), participating ref-based metrics (middle), and participating QE metrics (bottom).

metrics. Ref-based QE metrics perform slightly better than QE metrics. Regarding the two surface-form matching metrics, character-level chrF performs much better than SENT-BLEU on AFF/NEG, Number and D/T test cases, although slightly worse on the other two categories. The performances of embedding-based metrics vary greatly across both ref-based and QE metrics.

4.2 Comparison across error categories

Embedding-based metrics perform well on AFF/NEG cases as we assumed, as most embedding-based metrics (both ref-based and QE) achieve medium to strong correlations with human ranking. However, regarding the other four categories on word-level crucial errors, performances of some embedding-based metrics deteriorate significantly and only few metrics manage to reach medium-level correlation.

5 Discussion

5.1 Number as A Tough Issue

One of the focuses of our challenge set is number. Numbers are dispersed, rich in format, and semantically similar, making metrics hard to grasp the exact meaning. To analyze how metrics perceive and score numbers, we further divide it into four sub-categories:

- Same Format (SAME): Good and bad exam-

Metric	SAME	DIFF	SWAP	SEP
STEN-BLEU	-0.908	-0.807	-0.333	-0.630
chrF	-0.333	-0.572	-0.286	0.210
BERTScore	0.632	-0.393	-0.476	-0.383
BLEURT-20	0.678	0.490	0.000	0.481
COMET-20	0.011	-0.559	-0.095	0.630
COMET-QE	-0.034	-0.159	0.000	0.531
Yisi-1	0.586	-0.379	-0.476	-0.309
Baseline Avg.	**0.090**	**-0.340**	**-0.238**	**0.076**
COMET-22	0.747	-0.103	-0.143	0.358
MATESE	-0.839	-0.710	-0.857	-0.333
MEE	-0.701	-0.876	-0.810	-0.160
MEE2	0.747	0.283	-0.571	0.086
MEE4	0.816	0.421	-0.571	0.012
metricx_xl_DA	0.954	0.862	0.190	0.605
metricx_xl_MQM	0.770	0.724	0.571	0.580
metricx_xxl_DA	0.977	0.903	0.762	0.531
metricx_xxl_MQM	0.931	0.890	0.905	0.728
MS-COMET-22	0.057	-0.103	-0.190	-0.012
UniTE	0.655	-0.103	-0.381	0.457
UniTE-ref	0.655	-0.076	-0.190	0.556
Ref-based Avg.	**0.481**	**0.176**	**-0.107**	**0.284**
COMETKiwi	0.425	-0.090	0.048	0.457
Cross-QE	0.218	0.145	-0.095	0.630
HWTSC-Teacher-Sim	0.632	0.503	0.000	0.630
HWTSC-TLM	0.448	0.393	0.238	0.556
KG-BERTScore	0.655	0.503	-0.143	0.630
MATESE-QE	-0.839	-0.821	-0.762	-0.457
MS-COMET-QE-22	-0.011	0.034	-0.095	0.259
REUSE	0.747	0.641	-0.048	0.605
UniTE-src	0.264	-0.241	-0.143	0.679
QE Avg.	**0.282**	**0.119**	**-0.111**	**0.443**

Table 5: Kendall's tau-like correlation results on our challenge set. The horizontal lines delimit baseline metrics (top), participating reference-based metrics (middle), and participating QE metrics (bottom).

ples use different numbers in the same format (e.g. 1 & 2; three & four).

- Different Format (DIFF): The good examples contain correct numbers in a different format as reference while the bad examples contains an incorrect number in the same format as reference (e.g. 1 & two; 1,000,000 & 1 million).
- Swapped Number (SWAP): When a sentence contains two or more numbers, we swap the numbers to generate the bad translation.
- Thousand Separator (SEP): Thousand separators are not required but help improve readability. Test sets under this category compare numbers without thousand separators with those in wrong formats (e.g. 1 000; 10,00; 1.000).

Table 5 presents results on the number subcategories. According to the table, surface-form matching metrics perform worse under all the four subcategories. Although embedding-based metrics in general perform much better, those metrics still perform worse under the SWAP subcategory.

5.1.1 Numeral vs. Number

In daily usage, there is no strict rule about when to use numerals or numbers. In some cases, numeral and number are just two different symbols to express the same meaning and as a result can be regarded as synonyms. According to table 5, the majority of embedding-based metrics perform relatively well on discerning differences among numerals or among numbers (SAME). To be more specific, if the good and adversarial examples contain different numbers in the same format, even if the format is different from that used in the reference, the possibility for metrics to discern between the correct and incorrect numbers is relatively high.

However, performances of these metrics under the DIFF category deteriorate to varying extents. In other words, if the good example contains a correct number in different format and the adversarial example contains an incorrect number but in the same format as that in the reference, metrics are likely to assign a higher score to the adversarial example.

The result demonstrates that contextual embeddings fail to relate semantically similar numbers and numerals. Instead, they seem to rely more on the "surface similarity". Another example to buttress this assumption is the metrics' performances in the thousand separator category, where there is about 50% of chance that metrics score numbers with wrong separator formats higher than those without separators.

Although neural machine translation models seldom translate numbers wrong, outputs do use different number formats. When these metrics are used to measure model performances, they incline to wrongly penalize sentences using a different number format, thus leading to unfair evaluations.

5.1.2 Does Number Difference Count?

We further conducted two experiments to examine if metrics' capability of distinguishing numbers improves when the difference between the correct and incorrect numbers turns greater. The sentence shown in Table 3 is used for the two experiments.

In the first experiment, we replace the number in the reference (REF in table 3) to its numeral format "thirty-four" and denote the sentence as good-translation x1. Then we replace the number in the reference to other Arabic numbers ranging from 1 to 100 to generate a set of comparative candidates denoted as bad-translations Y{y_1,y_2,...y_{100}}.

In the second experiment, we denote another correct post-edit result as good-translation x2 (GOOD in table 3), and alter the numeral in x2 to Arabic numbers ranging from 1 to 100 (denoted as bad-translations Z{z_1,z_2,..., z_{100}}).

We calculated BERTScore of x1, x2, Y and Z against the reference and the result is presented in figure 1. When there is no other difference between reference and candidates except the number, it seems easier for BERTScore and BLEURT to discern number differences even in different formats. In addition, as the difference between numbers becomes greater, the gap of scores expands. However, when there are other differences between the reference and candidates, it becomes harder for BERTScore to quantify the error, as BERTScore gives the majority of candidates in Z higher scores than x2. And greater difference between numbers seems not help. However, BLEURT remains a good performance in the second experiment, which is consistent with our challenge test results.

5.1.3 Do Metrics Understand Number?

Another interesting finding regarding number is that all metrics perform badly under the SWAP category (only three ref-based metrics managed to achieve medium-level correlation). Swapping two numbers in a sentence causes drastic changes in meaning but metrics lack the capability to identify such changes.

5.2 Is Source/Ref Information Helpful?

In general, QE metrics perform relatively worse than ref-based metrics, but the gap is smaller than we assumed. By just leveraging source-side information, the average of QE metrics almost reaches medium-level correlation. This gives rise to a question: if a metric leverages both source-side and target-side information, will the accuracy improve?

The implementations of COMET-22 and COMET-Kiwi are almost the same but COMET-22 leverages both source-side and target-side text while COMET-Kiwi uses only source-side text. When we compare the performances of the two metrics, we find that COMET-22 outperforms COMET-Kiwi under the Number and D/T categories. However, COMET-Kiwi outperforms COMET-22 under the NE&Term, Unit and AFF/NEG categories.

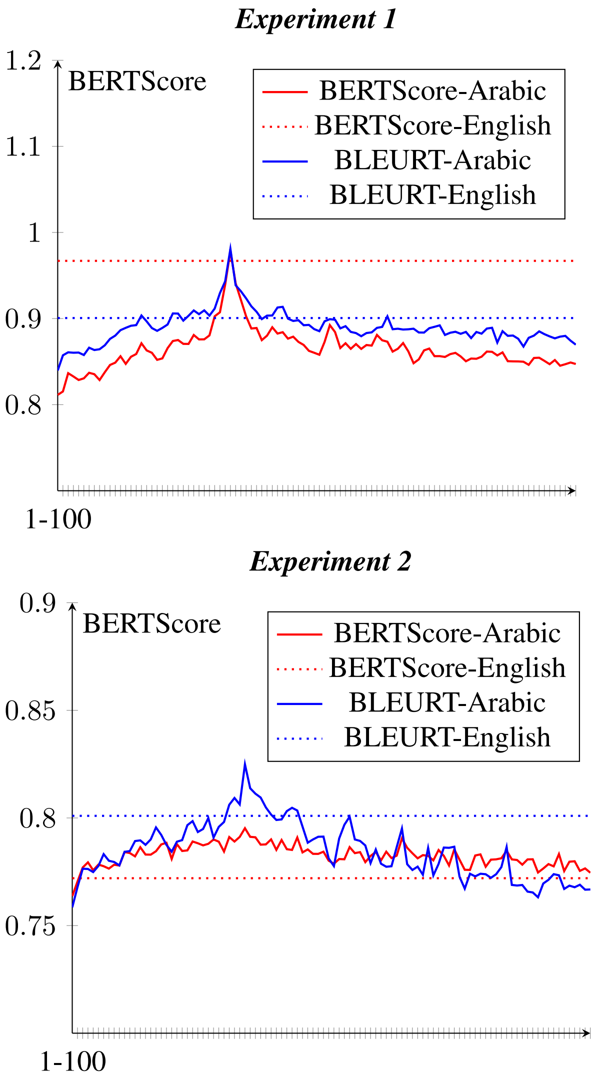


Figure 1: Results for experiment 1 and 2. The dotted lines indicate the scores for x1 and x2, while the solid lines represent the results of Y and Z.

The result indicates that while in some cases, reference-side information helps improve accuracy; in other cases, reference-side information surprisingly causes performance deterioration.

Among all the participating ref-based metrics, although some leverage source-side information while the others do not, their implementations vary. So we are unable to draw a conclusion that whether adding source-side information to a ref-based metric helps improve accuracy. More ablation experiments are required.

5.3 Synonym is Still A Tough Issue

Although embedding-based metrics claim better capture of synonyms, the result shows that there is still a long way to go. Not only numbers, test cases under NE&Term, D/T, and Unit categories all aim at examining metrics' ability to relate different formats of words that express the same meaning. The results show that metric performances vary greatly under these categories. The variations demonstrate that there should be a solution to this problem. However, at the time of writing, we have no detailed information about the implementations of those well-performed metrics. For more details, please refer to WMT Metric summary report and their system reports.

Metric	FLORES	WMT
BLEU	-0.728	-0.705
chrF	-0.398	-0.386
BERTScore	-0.073	-0.327
BLEURT-20	0.529	0.457
COMET-20	-0.042	-0.233
Yisi-1	-0.016	-0.280
COMET-QE	0.225	-0.062
Baseline Avg.	**-0.072**	**-0.220**
COMET-22	0.450	0.198
MATESE	-0.492	-0.457
MEE	-0.644	-0.693
MEE2	0.047	0.074
MEE4	0.178	0.162
metricx_xl_DA$_2$019	0.801	0.752
metricx_xl_MQM$_2$019	0.796	0.764
metricx_xxl_DA	0.848	0.794
metricx_xxl_MQM	0.911	0.823
MS-COMET-22	0.084	-0.068
UniTE	0.398	0.162
UniTE-ref	0.435	0.239
Ref-based Avg.	**0.318**	**0.229**
COMETKiwi	0.450	0.209
Cross-QE	0.445	0.221
HWTSC-Teacher-Sim	0.450	0.440
HWTSC-TLM	0.393	0.392
KG-BERTScore	0.487	0.398
MATESE-QE	-0.649	-0.705
MS-COMET-QE-22	0.215	0.068
REUSE	0.571	0.481
UniTE-src	0.335	0.192
QE Avg.	**0.300**	**0.188**

Table 6: A comparison of metric performances on Flores and WMT test cases. The horizontal line delimit baseline metrics (top) and participating reference-based metrics (bottom).

5.4 Do Metrics Suffer from Domain Issue?

We build our challenge set based on two open-source test sets: Flores 101 and WMT21 Zh-En. Hanna and Bojar (2021) claim that when the reference is a post-edit, BERTScore performs poorly as the post-edit may have high lexical overlap with machine translations. In our experiment setting, the candidate sentences are post-edits, which are stylistically similar to references in the WMT21 Zh-En news test sets, as the references are translations provided by professional translators. On the contrary, the Flores 101 En-Zh test set is translated from English to Chinese, so the English side is original and less stylistically similar to post-edits.

We calculate each metric's performance on Flores and WMT test cases (see table 6). Surface-form matching metrics are least influenced by the difference. For both ref-based and ref-free metrics, while some metrics (e.g. Metricx, HWTSC-Teacher-Sim) remain almost same performance on cases from the two sources, some metrics (e.g. COMET-22, UniTE) perform far worse on WMT cases.

The result shows that the generalizability of metrics varies. While good metrics can remain the same performance on test sets in different domains and of different styles, some metrics suffer greatly from domain issues. We assume the reasons for such performance gaps including: 1) WMT test cases are longer than Flores cases in average, making the cases harder to score; 2) big data for training pre-trained models are mostly native monolinguals so these models are better at encoding native languages than "translationese". However, more ablation experiments are required and generalizability should be concerned when developing metrics.

6 Conclusion

This paper presents our submitted challenge set to the WMT22 Metrics Challenge Sets Subtask and various metrics' performances on our set. Our set focuses on five categories of errors and the result shows that while most metrics are able to identify catastrophic sentence-level affirmation/negation errors, some metrics fail at discerning word-level keyword errors and capturing synonyms of such words. The results show that references are not always useful for a metric to identify errors. In addition, generalizability of metrics should be considered as some metrics are susceptible to test sets styles. The majority of metrics fail to meet the requirements (Banerjee and Lavie, 2005; Koehn,

2009) we discuss in the introduction section. They fail to identify nuances in quality and provide reliable scores, and suffer from domain issues as well.

The limitation of this research is that all of the perturbations are human-crafted, and these errors may seldom occur in neural machine translations. To further analyze metric performance in real settings, we will try to annotate and categorize real machine translation errors and evaluate metric performance accordingly.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Michael Hanna and Ondřej Bojar. 2021. A fine-grained analysis of BERTScore. In *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517, Online. Association for Computational Linguistics.

Philipp Koehn. 2009. *Statistical Machine Translation*. Cambridge University Press.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Chi-kiu Lo. 2020. Extended study on using pretrained language models and YiSi-1 for machine translation evaluation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 895–902, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Ricardo Rei, Ana C Farinha, Chrysoula Zerva, Daan van Stigt, Craig Stewart, Pedro Ramos, Taisiya Glushkova, André F. T. Martins, and Alon Lavie. 2021. Are references really needed? unbabel-IST 2021 submission for the metrics shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1030–1040, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020a. Comet: A neural framework for mt evaluation. *arXiv preprint arXiv:2009.09025*.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020b. Unbabel's participation in the wmt20 metrics shared task. *arXiv preprint arXiv:2010.15535*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020a. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Thibault Sellam, Amy Pu, Hyung Won Chung, Sebastian Gehrmann, Qijun Tan, Markus Freitag, Dipanjan Das, and Ankur Parikh. 2020b. Learning to evaluate translation beyond English: BLEURT submissions to the WMT metrics 2020 shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 921–927, Online. Association for Computational Linguistics.

Aaron Smith, Christian Hardmeier, and Joerg Tiedemann. 2016. Climbing mont BLEU: The strange world of reachable high-BLEU translations. In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, pages 269–281.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

A Appendix

SRC:	死亡人数至少为***15*** 人，预计还会增加。
REF:	The death toll is at least ***15***, a figure which is expected to rise.
GOOD:	The death toll is at least ***fifteen*** and is expected to rise.
BAD:	The death toll is at least ***fourteen*** and is expected to rise.
Phenom:	Number (Same Format)
Source:	Flores

SRC:	湖南红色旅游文化节已成功举办***16***届，是全国红色旅游的知名品牌。
REF:	The Hunan Red Tourism and Culture Festival has been successfully held for ***16*** years, making it a famous red tourism brand in China.
GOOD:	Hunan Red Tourism Culture Festival has been successfully held for ***sixteen*** times and is a well-known brand of red tourism in China.
BAD:	Hunan Red Tourism Culture Festival has been successfully held for ***14*** times and is a well-known brand of red tourism in China.
Phenom:	Number (Different Format)
Source:	WMT

SRC:	投票存在两极分化的情况，***29%*** 的受访者认为澳大利亚应该尽快成立共和国，***31%*** 的人则认为澳大利亚永远不应该成立共和国。
REF:	At the extremes of the poll, ***29 per cent*** of those surveyed believe Australia should become a republic as soon as possible, while ***31 per cent*** believe Australia should never become a republic.
GOOD:	The vote was polarised, with ***29%*** of respondents saying Australia should become a republic as soon as possible and ***31%*** saying it should never become a republic.
BAD:	The vote was polarised, with ***31%*** of respondents saying Australia should become a republic as soon as possible and ***29%*** saying it should never become a republic.
Phenom:	Number (Swapped Number)
Source:	Flores

SRC:	除了大件，让傅昆宝两口子头疼的还有家里***1000***多斤粮食和新买的一些家具。
REF:	Apart from the large items, the over ***1,000*** jin (500 kg) of grain and newly bought furniture was also a headache Fu Kunbao and his wife.
GOOD:	In addition to big items, Fu Baokun and his wife don't know how to deal with more than ***1000*** jin of grain and some newly bought furniture in the home.
BAD:	In addition to big items, Fu Baokun and his wife don't know how to deal with more than ***1.000*** Jin of grain and some newly bought furniture in the home.
Phenom:	Number (Thousand Separator)
Source:	WMT

SRC:	“我是***7***月***7***日来北京的，当时其实有点担心疫情，还提前三天做了核酸检测，是带着酒精棉和检测报告来布展的。”
REF:	“I arrived in Beijing on ***July 7***, and at the time I was a little worried about the pandemic, so took the nucleic acid test three days in advance, and I came here with alcohol pads and my test report. ”
GOOD:	"I arrived in Beijing on ***the 7th of July***. At that time, I was a little worried about the pandemic so I did a nucleic acid test three days in advance, and I took alcohol pads and the test report to set up the exhibition."
BAD:	"I arrived in Beijing on ***June 7***. At that time, I was a little worried about the pandemic so I did a nucleic acid test three days in advance, and I took alcohol pads and a test report to set up the exhibition."
Phenom:	Date & Time
Source:	WMT

SRC:	美国地质调查局国际地震地图显示，冰岛在前一周并未发生地震。
REF:	The ***United States*** Geological Survey international earthquake map showed no earthquakes in Iceland in the week prior.
GOOD:	The ***U.S.*** Geological Survey International Earthquake Map shows no earthquakes in Iceland in the previous week.
BAD:	The ***United Kingdom*** Geological Survey International Earthquake Map shows no earthquakes in Iceland in the previous week.
Phenom:	Named Entity & Terminology
Source:	Flores

SRC:	到今天早些时候，风速为每小时***83*** 公里左右，预计会不断减弱。
REF:	By early today, winds were around 83 ***km/h***, and it was expect to keep weakening.
GOOD:	By early today, the wind speed was about 83 ***kilometers per hour***, and it is expected to continue to weaken.
BAD:	By early today, the wind speed was about 83 ***m/h***, and it is expected to continue to weaken.
Phenom:	Unit Format
Source:	Flores

SRC:	不久前，他在布里斯班公开赛上败于拉奥尼奇。
REF:	He ***recently*** lost against Raonic in the Brisbane Open.
GOOD:	***Not long ago***, he lost against Raonic at the Brisbane International tournament.
BAD:	***Long ago***, he lost against Raonic at the Brisbane International tournament.
Phenom:	Unit Format
Source:	Flores

MS-COMET: More and Better Human Judgements Improve Metric Performance

Tom Kocmi **Hitokazu Matsushita** **Christian Federmann**
Microsoft, 1 Microsoft Way, Redmond, WA 98052, USA
{tomkocmi,himatsus,chrife}@microsoft.com

Abstract

We develop two new metrics that build on top of the COMET architecture. The main contribution is collecting a ten-times larger corpus of human judgements than COMET and investigating how to filter out problematic human judgements. We propose filtering human judgements where human reference is statistically worse than machine translation. Furthermore, we average scores of all equal segments evaluated multiple times. The results comparing automatic metrics on source-based DA and MQM-style human judgement show state-of-the-art performance on a system-level pair-wise system ranking. We release both of our metrics for public use.[1]

1 Introduction

Automatic metrics for machine translation (MT) evaluation are commonly used as the primary tool for comparing the translation quality of MT systems, often without evaluating systems with the human judgement that can be expensive and time-consuming (Marie et al., 2021). Therefore, studying and developing metrics that correlate well with human judgement is critical.

There is an increasing effort in the evaluation of automatic MT metrics, leading with the annual evaluation of metrics at the WMT conference (Freitag et al., 2021b,a; Kocmi et al., 2021; Mathur et al., 2020b). Most research has focused on comparing segment-level or system-level correlations between absolute metric scores and human judgements. However, Mathur et al. (2020a) emphasize that this scenario is not identical to the everyday use of metrics, where instead, researchers and practitioners use automatic scores to compare pairs of systems. For example, when claiming a new state-of-the-art, evaluating different model architectures, and deciding whether to publish results or deploy new production systems.

In this work, we focus on training automatic metric based on COMET architecture (Rei et al., 2020) utilizing a large internal trainset of human segment-level judgements. Additionally, we evaluate the metrics in a pair-wise system-level evaluation against human judgement.

We develop two metrics: *MS-COMET* intended for reference-based evaluating systems, while *MS-COMET-QE* is designed for quality estimation or source-based evaluation. We use the suffix "-22" to differentiate the models from potential future releases.

2 Related work

There are two main categories of automatic MT metrics: (1) string-based metrics and (2) metrics using pretrained models. The former compares the coverage of various substrings between the human-generated reference and MT translations, this group includes metrics such as ChrF (Popović, 2015), BLEU (Papineni et al., 2002), or TER (Snover et al., 2006). String-based methods largely depend on the quality of reference translations. However, their advantage is that their performance is predictable as it can easily diagnose which substrings affect the score the most.

The latter category of pretrained methods consists of metrics that usually use pretrained models to evaluate the quality of MT translations given the source sentence, the human reference, or both. Evaluation metrics from this category includes COMET (Rei et al., 2020), BLEURT (Sellam et al., 2020), or BERTScore (Zhang* et al., 2020). They are not strictly dependent on the reference quality (for example, they can better evaluate synonyms or paraphrases), and many studies (Freitag et al., 2021b; Mathur et al., 2020b; Kocmi et al., 2021) showed their superiority over string-based metrics. On the other hand, their performance is influenced

[1] https://github.com/MicrosoftTranslator/MS-Comet

by the data on which they have been trained, which may introduce bias, and the pretrained models present a black-box problem where it is challenging to diagnose potential unexpected behavior of the metric.

A separate category of automatic metrics is whether they need a human reference for evaluation. Automatic metrics that calculate scores without the need for reference (quality estimation) open the possibility of evaluating monolingual testsets that can be tailored for a specific domain without the need to build expensive human references.

We build our metric with the architecture of COMET (Rei et al., 2020).[2] It uses the Estimator model which uses pretrained language models XLM-RoBERTa to encode source, MT hypothesis and reference in the same cross-lingual space. The model is then fine-tuned on human judgement data. We use the identical hyper-parameters as COMET.

3 Human Judgement Trainset

For training our models, we use a mix of public and internal data that we further denoise by filtering out potentially problematic human judgements.

We use the same human judgments data used to train the COMET model, i.e. WMT 2017-2019 (Barrault et al., 2019; Bojar et al., 2018, 2017). To test the quality of metrics, we use WMT 2020 (Barrault et al., 2020), WMT 2021 (Akhbardeh et al., 2021) and MQM 2021 (Freitag et al., 2021b). Furthermore, we submitted our model to WMT Metrics Shared Task 2022.

In addition to publicly available data, we use a set of internal data, described in Kocmi et al. (2021) plus newer data collected over the last year. All our internal data are collected with the use of expert annotators. We use a mix of human judgement methods: source-based Direct Assessment (srcDA) (Graham et al., 2013; Federmann, 2018), contrastive Direct Assessment (contrDA, which asks users to rate pairs of system outputs), and SQM presented at WMT General MT 2022 (which uses labeled scale). All collected labels are on a scale of 0-100, where the interface structure is the main difference for human annotators.

We use internal testsets for human judgements that have been translated with a tandem of two professional translators, following findings of Freitag et al. (2020) that high-quality reference plays an essential role in automatic evaluation.

	Langs.	Domains	Segments
All available data			6.53 M
Removed low-quality			0.79 M
Removed WMT refDA			0.35 M
Removed by averaging			2.12 M
MS-COMET	111	15	2.06 M
MS-COMET-QE	113	15	3.43 M
COMET	13	1	0.66 M

Table 1: The statistics of the training corpora and the effect of filtering in terms of unique languages on the target side, unique domains, and count of training segments used to train MS-COMET, MS-COMET-QE, and original COMET.

In contrast to publicly available data that uses only the News domain, we use a mix of fifteen domains (news, conversation, legal, medical, social, e-commerce, tech, finance, and others). The news domain is the largest domain utilizing at least half of human judgements. Our collection of human judgement data covers 113 languages in contrast to 13 on which COMET is trained. A complete list of all supported languages and counts of human judgement for the largest translation directions are in Appendix A.

Reference-less metric MS-COMET-QE is trained using all training data and removing reference translations. Additionally, many human judgments are evaluated on data that are missing human reference, which is the reason for having more training data for MS-COMET-QE.

3.1 Using raw scores instead of z-scores

The z-score has been introduced (Graham et al., 2013) to resolve an issue with different strategies annotators may apply when judging systems. For example, an overly strict annotator may harshly penalize a system from which he annotated more segments. We partly avoid this problem in our data via a better sampling technique. We sample uniformly from each evaluated system in a way that each annotator evaluates the same number of sentences from each system. Therefore, different strategies should penalize all systems similarly.

As Knowles (2021) pointed out that z-score standardization of human judgements normalizes away both inter-annotator and system quality differences, and since we do not have a mechanism to avoid normalizing away system quality differences. There-

[2] To differentiate models, we are going to use COMET to reference Rei et al. (2020) models from ours labeled as MS-COMET

fore, we decided to use raw scores (0-100) instead of the z-score standardized counterpart.

Using raw scores has the benefit that it gives final scores some meaning. For low-quality languages, we may expect scores in the lower range (0-50), while for high-quality languages, the scores generally can be higher. Z-scores only do not represent any meaning. However, we do not advocate using our metric in an absolute fashion or comparing quality across languages.

However, we want to point out that we have seen only minor improvement when training metrics using raw scores in contrast to z-scores. Therefore, this decision is mainly on a pragmatic layer.

3.2 Professional annotators only

Freitag et al. (2021a) discuss that the quality of crowd-based human judgement is suboptimal, and human evaluation should focus on expert annotators. Professional annotators collect our internal human labels. However, data from WMT are collected in two different setups when one uses crowd-workers.

The language pairs that are from English or not containing English are collected with semi-professional to professional annotators and using source-based DA, which avoids reference bias. On the other hand, all into English language pairs are collected with crowd-workers with reference-based DA. For this reason, we decided to remove all WMT reference-based DA human judgement from our datasets, and therefore, we use only internal into-English human assessments.

3.3 Averaging same human judgement

In our data, many human judgment campaigns evaluate identical triplets *(source, hypothesis, reference)* in different campaigns. This happens when we compare identical baseline system across different campaigns or when a candidate system from the earlier campaign is later evaluated as a baseline system.

We notice that human scores fluctuate every time each triplet is evaluated. We have decided to average scores for all identical triplets to normalize the noise and balance the trainset. Averaging equal scores improved the performance of the metric.

We also experimented with taking a median of the scores, but the results have been a bit worse than averaging.

3.4 Removing low-quality human judgements

In our human annotation campaigns, we often include human reference translation as another system to measure how close MT systems are to human reference. However, scoring human references can also be used as a sanity check for the quality of campaigns or human references. Whenever we see a campaign where human reference is worse than the MT system, it suggests one of the following three scenarios: human reference contains error translations, human judgement is too noisy or misleading, or the MT system performs better than human translators. If we assume that MT systems are not outperforming human translators, a lower human reference score suggests either broken reference translation or a noisy campaign. Neither of these two outcomes is desirable for fine-tuning automatic metrics.

Therefore, we remove all campaigns containing human reference as an additional system, where any of the systems is statistically significantly better than human translation under the Mann–Whitney U test and alpha threshold of 5%.

4 Evaluation

Evaluation of automatic metrics is a challenging task investigated in a yearly WMT Metrics shared task (Freitag et al., 2021b). However, there is no community-agreed testset or evaluation method for comparing with humans that are considered gold standards.

There are different dimensions how to evaluate automatic MT metrics. Let's summarize the main differing points:

- **Human annotation methods** - source-based direct assessment (DA) (Graham et al., 2013), reference-based DA (Graham et al., 2013), contrastive DA (Akhbardeh et al., 2021), Multidimensional Quality Metrics (MQM) (Freitag et al., 2021a)

- **Granularity of evaluation** - evaluating correlation with human on a segment-level or system-level

- **Correlation method** - correlation of absolute values (Pearson or Kendall-like, Mathur et al., 2020b) or correlations in pairwise approach (pairwise accuracy, Kocmi et al., 2021; Mathur et al., 2020a

- **Usage of unlabeled part of testset** - human judgment often evaluates only a subset of the testset. Metrics can use the remaining unlabelled segments (especially for system-level setup)

- **Normalize human behavior** - use raw human scores or normalize them with z-score standardization

- **Evaluating human reference** - if additional human translated references should be evaluated like one of the systems (Freitag et al., 2021b)

- **Evaluating outlier systems** - absolute value correlations via Pearson are sensitive to outliers, therefore Mathur et al. (2020a) recommends removing outlier systems from evaluation.

The list is incomplete as there are other nuances, such as removing outlier systems, using only statistically significant pairs of systems, underlying quality of human judgement, etc.

Evaluating all combinations of approaches is not reasonable. Therefore we mainly follow the approach defined by Kocmi et al. (2021) and also used by WMT Metrics 2021 (Freitag et al., 2021b). Here is a list of constraints for the evaluation:

- We use only testsets produced by professional annotators as described in Section 3.2. Thus, we do not evaluate over reference-based DA.

- We focus on a system-level pairwise setup as the important use-case for automatic metrics (Kocmi et al., 2021). Thus we do not evaluate absolute value correlations with humans. Furthermore, this avoids the problem with outlier systems.

- We use only segments that have been evaluated by humans (unlabelled segments of testsets are not used).

- We use z-score normalization mainly to be comparable with past work. However, we do not consider z-score as a good standardization approach.

- We do not remove additional human references from the evaluation as metrics should be able to evaluate any translation (not only those produced with current MT systems).

	LPs	System pairs	Method
WMT20	8	565	srcDA
WMT21	9	1000	srcDA
WMT21-contr	3	198	contrDA
MQM21-news	3	301	MQM
MQM21-ted	3	247	MQM

Table 2: The statistics of human judgement sets are used for testing automatic metrics.

4.1 Evaluation methodology

We use system-level pairwise accuracy as introduced by Kocmi et al. (2021), which evaluates how often metric agrees on the ranking of two systems with human rank:

$$\text{Accuracy} = \frac{|\text{sign}(\text{metric}\Delta) \;=\; \text{sign}(\text{human}\Delta)|}{|\text{all system pairs}|}$$

We use implementation by Freitag et al. (2021b); therefore, results on the MQM21 testset agree with their findings. We use bootstrap resampling to calculate which metrics are not significantly outperformed by the winning metric with an alpha threshold of 0.05.

To test automatic metrics, we use publicly available data from different sources. You can find statistics in Table 2.

- **WMT20** and **WMT21** - we use source-based DA from Barrault et al. (2020) and Akhbardeh et al. (2021)

- **WMT21-contr** - we use contrastive DA from Akhbardeh et al. (2021). This is the only source of truly pairwise human judgements, where annotators see the outputs of two systems next to each other. We collect those pairs of systems evaluated to each other.

- **MQM21-news** and **MQM21-ted**- we MQM data from Freitag et al. (2021b), both testsets evaluate same set of systems but over different domains.

Additionally, we combine **all** testsets to calculate pairwise accuracy across all system pairs, simply by counting all system pairs where the metric agrees with human overall evaluated system pairs in all testsets.

4.2 Evaluated automatic metrics

We train two metrics MS-COMET trained with human-produced references and MS-COMET-QE

	All	WMT20	WMT21	WMT21-contr	MQM21-news	MQM21-ted
n	2311 ↓	565	1000	198	301	247
MS-COMET-22	**0.826 (1)**	**0.892 (1)**	**0.864 (1)**	0.722 (2)	0.714 (4)	**0.745 (3)**
MS-COMET-QE-22	**0.821 (2)**	0.873 (2)	0.847 (2)	**0.808 (1)**	0.734 (2)	0.713 (6)
Bleurt	**0.820 (3)**	0.869 (3)	**0.864 (1)**	0.702 (3)	0.718 (3)	**0.749 (2)**
COMET	0.816 (4)	0.869 (3)	**0.864 (1)**	0.677 (5)	0.678 (5)	**0.781 (1)**
COMET-QE	0.800 (5)	0.848 (6)	0.839 (3)	0.692 (4)	**0.774 (1)**	0.652 (7)
BERTScore	0.790 (6)	0.853 (5)	0.836 (4)	0.722 (2)	0.621 (6)	0.721 (5)
chrF	0.770 (7)	0.857 (4)	0.793 (5)	0.702 (3)	0.621 (6)	0.713 (6)
BLEU	0.688 (8)	0.848 (6)	0.622 (7)	0.601 (7)	0.618 (7)	0.741 (4)
TER	0.669 (9)	0.766 (7)	0.657 (6)	0.616 (6)	0.585 (8)	0.636 (8)

Table 3: The main results for pairwise accuracy in a system-level setting. The bold scores represent metrics that are not statistically different from the winning metric with a 0.05 alpha level. The numbers in brackets show the rank of metrics. The "n" represents the number of system pairs in each evaluation.

trained only with sources and MT hypothesis. We use identical hyper-parameters as the original COMET model (Rei et al., 2020), and the models are trained for precisely four epochs.

We compare our metrics to publicly available metrics, and either have the highest correlation with humans - COMET, BLEURT, and BERTScore (Kocmi et al., 2021; Freitag et al., 2021a) or are widely used in MT field (BLEU, ChrF, TER). We use default parameters and models for each of them, specifically:

For BLEU (Papineni et al., 2002), ChrF (Popović, 2015), and TER (Snover et al., 2006), we use SacreBLEU implementation https://github.com/mjpost/sacrebleu/ (Post, 2018) version 2.0.1. We use the "mteval-v13a" tokenizer for all language pairs except for Chinese and Japanese, which use their separate tokenizer, as is recommended.

For BERTScore (Zhang* et al., 2020), we use https://github.com/Tiiiger/bert_score version 0.3.11.

For BLEURT (Sellam et al., 2020), we use the recommended model "bleurt-20" and implementation https://github.com/google-research/bleurt.

For COMET (Rei et al., 2020), we use recommended model "wmt20-comet-da" and for COMET-QE we use "wmt21-comet-qe-mqm". The implementation is https://github.com/Unbabel/COMET in version 1.1.0.

5 Results

The results for the pairwise system-level scenario are in Table 3. The results over 2311 system pairs show that both our metrics outperform all other state-of-the-art metrics, with only Bleurt not being statistically worse than our metrics.

n	23595
MS-COMET-QE-22	**0.597 (1)**
COMET-QE	**0.596 (2)**
MS-COMET-22	**0.594 (3)**
Bleurt	**0.593 (4)**
COMET	0.586 (5)
BERTScore	0.567 (6)
chrF	0.557 (7)
TER	0.536 (8)
sentBLEU	0.535 (9)

Table 4: The results for pairwise accuracy in a segment-level setting over *WMT21-contr* testset. The "n" represent a number of segment pairs used in the evaluation.

The results over individual testsets show that our metrics are ranked among the top-performing metrics. Interestingly, *MQM21-news* domain seems to be easier for Quality Estimation metrics, while *MQM21-ted* shows the opposite direction. These results are interesting as the underlying systems are identical except for additional human reference.

Lastly, our metrics win in the *WMT21-contr* testset. This is the only genuinely pairwise testset where annotators saw systems next to each other while evaluating them.

Although we focus on a system-level evaluation, we evaluate how metrics perform in a segment-level setting for completeness. We use the testset *WMT21-contr* to calculate accuracies in the same fashion as for system-level scenario, but taking pairs of segment annotations instead of system-level scores. The segment-level results in Table 4 show that our metrics, COMET-QE, and Bleurt are in the winning cluster outperforming other metrics.

6 Conclusion

We have investigated the training COMET model with a larger corpus of human judgements covering multiple domains and 113 languages.

We employed several steps of filtering low-quality or repetitive human judgement.

With those data, we trained two metrics: MS-COMET-22 and MS-COMET-QE-22, that outperform other current MT metrics on a pair-wise system-level decision task.

We release the metrics for public use.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Christian Federmann. 2018. Appraise evaluation framework for machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 86–88, Santa Fe, New Mexico. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021a. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be guilty but references are not innocent. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 61–71, Online. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021b. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Rebecca Knowles. 2021. On the stability of system rankings at WMT. In *Proceedings of the Sixth Conference on Machine Translation*, pages 464–477, Online. Association for Computational Linguistics.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. 2021. Scientific credibility of machine translation research: A meta-evaluation of 769 papers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7297–7306, Online. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020a. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020b. Results of the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A List of languages

Our collection of human judgements covers 113 languages, language variants, or writing systems. Here is the complete list. Note that XLM-Roberta does not support some languages:

Afrikaans, Albanian, Amharic, Arabic, Armenian, Assamese, Azeri, Bangla, Bashkir, Basque, Bosnian, Bulgarian, Burmese, Catalan, Central Kurdish, Chinese (Literary), Chinese (People's Republic of China), Chinese (Taiwan), Chinese Yue, Chuvash, Classic Chinese (Simplified), Croatian, Czech, Danish, Dari, Divehi, Dutch, English, Estonian, Faroese, Fijian, Filipino, Finnish, French, French (Canada), Galician, Georgian, German, Greek, Gujarati, Haitian Creole, Hebrew, Hindi, Hmong, Hungarian, Icelandic, Indonesian, Inuktitut, Inuktitut (Latin), Inuinnaqtun, Irish, isiZulu, Italian, Japanese, Kannada, Kazakh, Khmer, Kiswahili, Korean, Kurdish, Kyrgyz, Lao, Latvian, Lithuanian, Macedonian, Malagasy, Malay, Malay Standard, Malayalam, Maltese, Maori, Marathi, Mongolian, Mongolian (Cyrillic), Nepali, Norwegian, Odia, Otomi, Pashto, Persian, Polish, Portuguese (Brazil), Portuguese (Portugal), Punjabi, Romanian, Russian, Samoan, Serbian (Cyrillic), Serbian (Latin), Slovak, Slovenian, Somali, Spanish, Swedish, Tahitian, Tajik, Tajiki, Tamil, Tatar, Telugu, Thai, Tibetan, Tigrinya, Tongan, Turkish, Turkmen, Ukrainian, Upper Sorbian, Urdu, Uyghur, Uzbek, Vietnamese, Welsh.

Furthermore, our human judgement data are not balanced. In some translation directions, we have more human-labeled data than in others. Table 5 shows the largest forty translation directions in our training data corpus.

	Mono	With ref
English - German	175k	103k
English - Chinese	117k	80k
English - Czech	93k	71k
English - Russian	92k	72k
English - French	66k	36k
Chinese - English	63k	33k
German - English	60k	28k
Japanese - English	57k	35k
English - Japanese	55k	30k
English - Spanish	54k	27k
English - Dutch	52k	27k
French - English	50k	32k
English - Italian	50k	24k
Spanish - English	48k	29k
English - Finnish	45k	38k
Italian - English	44k	25k
English - Polish	43k	23k
Korean - English	39k	25k
English - Portuguese	38k	22k
English - Turkish	37k	24k
English - Korean	36k	20k
Polish - English	35k	19k
Czech - English	35k	18k
English - Hindi	34k	18k
English - Arabic	34k	17k
Dutch - English	33k	19k
Arabic - English	32k	16k
Russian - English	32k	17k
English - Estonian	28k	21k
English - Lithuanian	27k	18k
Hindi - English	25k	15k
Greek - English	25k	15k
English - Swedish	24k	13k
Turkish - English	23k	14k
English - Danish	21k	11k
Portuguese - English	21k	12k
English - Romanian	21k	14k
Swedish - English	21k	8k
Romanian - English	21k	16k
English - Slovak	21k	12k

Table 5: The number of human judgement for the forty largest translation directions. The counts represent data on the final filtered training set, where “Mono” are dataset counts for MS-COMET-QE and “With ref” are for MS-COMET.

Partial Could Be Better Than Whole. HW-TSC 2022 Submission for the Metrics Shared Task

Yilun Liu*, **Xiaosong Qiao***, **Zhanglin Wu, Chang Su, Min Zhang,**
Yanqing Zhao, Song Peng, Shimin Tao, Hao Yang, Ying Qin,
Jiaxin Guo, Minghan Wang, Yinglu Li, Peng Li, Xiaofeng Zhao
Huawei Translation Services Center, Beijing, China
{liuyilun3, qiaoxiaosong, wuzhanglin2, suchang8, zhangmin186, zhaoyanqing,
pengsong2, taoshimin, yanghao30, qinying, guojiaxin, wangminghan} @huawei.com

Abstract

In this paper, we present the contribution of HW-TSC to WMT 2022 Metrics Shared Task. We propose one reference-based metric, HWTSC-EE-BERTScore*, and four reference-free metrics including HWTSC-Teacher-Sim, HWTSC-TLM, KG-BERTScore and CROSS-QE. Among these metrics, HWTSC-Teacher-Sim and CROSS-QE are supervised, whereas HWTSC-EE-BERTScore*, HWTSC-TLM and KG-BERTScore are unsupervised. We use these metrics in the segment-level and system-level tracks. Overall, our systems achieve strong results for all language pairs on previous test sets and a new state-of-the-art in many sys-level case sets.

1 Introduction

Due to the expensive cost of manual evaluation, automatically evaluating the outputs of translation systems is critically important in the field of machine translation (MT) (Freitag et al., 2021a). Therefore, a lot of automatic metrics have been proposed to approach this task. According to whether the reference sentences are required or not, the metrics are categorized into two classes: (1) reference-based metrics like BLEU (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007), BERTScore (Zhang et al., 2020) and BLEURT (Sellam et al., 2020), which evaluate the hypothesis by referring to the golden reference; (2) reference-free metrics like YiSi-2 (Lo, 2019) and COMET-QE (Rei et al., 2020, 2021), which are also referred as quality estimation (QE). These metrics estimate the quality of hypothesis only based one source sentences without using references.

In this paper, we present the contribution of HW-TSC to the WMT 2022 Shared Task on Metrics. We participated in the segment-level and system-level tracks with 1 reference-based metric (HWTSC-EE-BERTScore*) and 4 reference-free metrics (HWTSC-Teacher-Sim, HWTSC-TLM, KG-BERTScore and CROSS-QE). Details of our metrics are illustrated in Table 1.

HWTSC-EE-BERTScore* (Entropy Enhanced Metrics)is built upon existing metrics, aiming to achieve a more balanced system-level rating by assigning weights to segment-level scores produced by backbone metrics. The weights are determined by the difficulty of a segment, which is related to the entropy of a hypothesis-reference pair. A translation hypothesis with a significantly high entropy value is considered difficult and receives a large weight in aggregation of EE-Metrics' system-level scores.

HWTSC-Teacher-Sim is a supervised reference-free metric with the framework of BERTScore (Zhang et al., 2020), which is obtained by fineturning the multilingual Sentence-BERT model (Reimers and Gurevych, 2019, 2020a). Both the unsupervised TearcherSim (Yang et al., 2022b,a) and the implicit multilingual word embedding alignment (Zhang et al., 2022b) have shown that the pretained multilingual Sentence-BERT model is very effective for both reference-based and reference-free MT evaluations on WMT DA (Direct Assessment) data. However, its performance on WMT MQM (Multidimensional Quality Metrics) data is poor. We propose an effective training strategy for the pretrained multilingual Sentence-BERT and a novel normalization method for the DA and MQM scores.

HWTSC-TLM (Zhang et al., 2022a) is an unsupervised reference-free metric which only uses the system translations as input and calculates the scores by a target-side language model. Although source sentences are not considered, the results of this metric with XLM-R (Conneau et al., 2020) on WMT19 are very promising.

KG-BERTScore (Wu et al., 2022) is an unsupervised reference-free metric, which incorporates multilingual knowledge graph into BERTScore

[*] equal contribution

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 549 - 557
December 7-8, 2022 ©2022 Association for Computational Linguistics

Metrics	Reference	Training	Segment-level	System-level
HWTSC-EE-BERTScore*	reference-based	unsupervised	✗	✓
HWTSC-Teacher-Sim	reference-free	supervised	✓	✓
HWTSC-TLM	reference-free	unsupervised	✓	✓
KG-BERTScore	reference-free	unsupervised	✓	✓
CROSS-QE	reference-free	supervised	✓	✓

Table 1: Description of 5 metrics participated in WMT 2022 Shared Task. ✓and ✗respectively indicate whether the metric participates the corresponding track or not.

(Zhang et al., 2020). The score of this metric is calculated by linearly combining the results of BERTScore and bilingual named entity matching.

CROSS-QE is an application of "QE as a metric". Based on our previous work (Yang et al., 2020; Wang et al., 2020; Chen et al., 2021), we propose a reference-free metric, like COMET-QE architecture.

2 Metrics

This section introduces our metrics for WMT Metrics 2022 Shared Task including Reference-based and reference-free.

2.1 Reference-based

This year, entropy-enhanced BERTScore (HWTSC-EE-BERTScore, or referred as EE-BERTScore in short) was used in the general tests of the system-level track. EE-BERTScore, built upon standard BERTScore (Zhang et al., 2019), is within one of the EE metrics proposed earlier (Liu et al., 2022). The main idea of EE metrics is to challenge the standard way of acquiring system-level scores that outputs a simple arithmetic average of scores on segments in the evaluation set, and to provide a framework that enhances existing MT metrics by assigning higher weights to the difficult samples in the evaluation set. The motivation is simple: for MT evaluation, it is not likely that human raters treat every source-reference pair equally. Those simple samples can be easily translated, leading to similar human scores given to different hypotheses, while the more challenging part in an evaluation set often distinguishes top candidates from inferior systems. Like different weights are assigned to questions in real-world examinations based on variant difficulties, MT evaluation metrics should also encourage systems that perform better on relatively difficult samples. In the preliminary experiment, we find that using only the difficult segments (usually counting for less than 5% of all segments in the whole evaluation set) to evaluate MT systems, doesn't lead the automatic metrics to give incorrect ratings for MT systems, and sometimes even improves the performances of metrics in terms of correlation with human DA scores. Thus, we proposed EE metrics, which emphasize the translation qualities of relatively difficult ones among all hypotheses given by a system and assign high weights to these hypotheses in the aggregation of system-level scores.

2.1.1 Working Process of EE Metrics

Currently, EE metrics determine the difficulty of a segment via the average qualities of hypotheses. The qualities are measured by the translation entropy (or chunk entropy) (Yu et al., 2015) between the reference and the hypothesis. For a human reference and a hypothesis given by an MT system, a high chunk entropy suggests high uncertainty of the translation (the more linguistically matched parts between the hypothesis and the reference is, the lower the uncertainty of the translation is) and a low entropy indicates good confidence of the given hypothesis in expressing the meaning of the source segment. For example, if a hypothesis is perfectly matched with a reference, then the entropy of the translation is zero, and if there is no matching token between the hypothesis and the reference, the chunk entropy is positive infinity, indicating a total uncertainty and disorderness of the translation.

Fig. 1 illustrates how EE metrics assign different weights to the segments in the evaluation set based on the computed entropy. Firstly, segments in the evaluation set are divided into two groups: easy samples and difficult samples. If the entropy of a hypothesis is higher than the threshold h, it is considered in the difficult group and vice versa. Then, hypotheses are assigned weights in the aggregation of final score based on the groups they belong to. Specifically, samples in the easy group receive a weight of w/N_e and samples in the difficult group

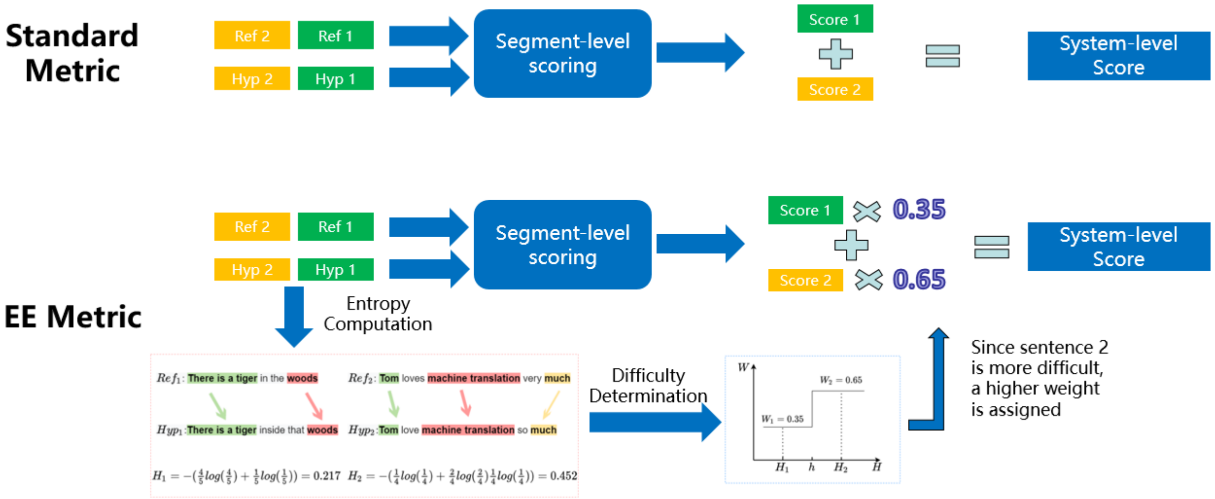

Standard Metric
Ref 2
Ref 1
Hyp 2
Hyp 1
Segment-level scoring
Score 1
Score 2
System-level Score
EE Metric
Ref 2
Ref 1
Hyp 2
Hyp 1
Segment-level scoring
Score 1
0.35
Score 2
0.65
System-level Score
Entropy Computation
Ref_1: There is a tiger in the woods
Ref_2: Tom loves machine translation very much
Hyp_1: There is a tiger inside that woods
Hyp_2: Tom love machine translation so much
$H_1 = -(\frac{4}{5}log(\frac{4}{5}) + \frac{1}{5}log(\frac{1}{5})) = 0.217$
$H_2 = -(\frac{1}{4}log(\frac{1}{4}) + \frac{2}{4}log(\frac{2}{4})\frac{1}{4}log(\frac{1}{4})) = 0.452$
Difficulty Determination
W
$W_1 = 0.35$
$W_2 = 0.65$
H_1
h
H_2
H
Since sentence 2 is more difficult, a higher weight is assigned

Figure 1: Workflow of EE metrics, assuming the evaluation set contains two segments with reference-hypothesis pairs (Hyp 1, Ref 1) and (Hyp 2, Ref 2).

receive a weight of $(1 - w)/N_d$, where N_e, N_d are the sizes of easy and difficult group, respectively, and w is a balance coefficient that, in our earlier version of EE metrics, may vary for different language pairs and evaluation datasets. Since the number of easy hypothesis is much larger than the number of difficult hypothesis for a given MT system, the weight of easy samples is much lower than the weight of difficult samples.

2.1.2 EE Metrics 2.0 vs. EE Metrics 1.0

The earlier version of EE metrics (denoted as EE metrics 1.0) has two hyper-parameters: h and w, involving in the selection of difficult samples and the determination of weights assigned to each group, respectively. The existence of such hyper-parameters hinders the application of EE metrics. What's worse, the hyper-parameters often alter for different language pairs and evaluation datasets (*e.g.*, we use up to 10 different parameters in our preliminary experiment, involving WMT 19 evaluation set), making it hard to estimate a feasible combination of parameters in the actual scenario. To alleviate such undesirable pain, we propose EE metrics 2.0 for this year's WMT metrics shared tasks. EE metrics 2.0 aims to reduce the hyper-parameters involved in the computation of system-level score as much as possible and offers a lightweight approach of computing weights for each segment. Specifically, EE metrics 2.0 doesn't require specifying h anymore, but automatically estimates thresholds based on a normal distribution fitting of average translation qualities (the average entropy) over all segments, aiming to find the threshold value of entropy where a sample has a significantly higher entropy than those of other samples in the datasets. Moreover, the estimation of w is simplified to a single value, instead of a series of different values for different language pairs. EE metrics 1.0 provides a formula to estimate w for every language pair, which is acquired based on the fitting of WMT 19 results. In contrast, the value of w doesn't change across different language pairs in EE metrics 2.0. Our submissions in WMT 2022 Metrics Shared Task contain three different configurations of values of w: 0.3, 0.5 and 0.8, which stand for different degrees of balance of weights received between difficult groups and easy groups.

2.2 Reference-free

In this section, we would introduce the four reference-free metrics.

2.2.1 HWTSC-Teacher-Sim

HWTSC-Teacher-Sim proposed by (Zhang et al., 2022b), is a Reference-free metric used for machine translation evalation by achieving cross-lingual word embedding alignment thrpugh multilingual knowledge distillation (MKD) (Reimers and Gurevych, 2020b). The procedure of multilingual knowleage distillation is described in the Figure 2. The teacher model is monolingual SBERT (Reimers and Gurevych, 2019) which achieves state-of-the-art performance for various sentence embedding tasks, and the student model is a multilingual pretrained model like mBERT or XLM-R before distillation. After MKD, the similarity score of sentence pairs in MT evaluation on the language

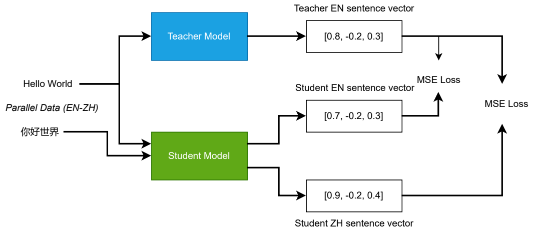


Figure 2: Multilingual knowledge distillation

model should be as high as possible. Based on this feature, embeddings of sentences are used to calculate the similarity score as a metric. And we achieve strong results using language models to calculate the similarity between sentence pairs in an supervised manner in MQM data.

2.2.2 HWTSC-TLM

HWTSC-TLM proposed by Zhang et al. (2022a) utilizes a pretrained multilingual model XLM-R (Conneau et al., 2020) to score the system translations, which is a zero-shot unsupervised metric for MT evaluation.

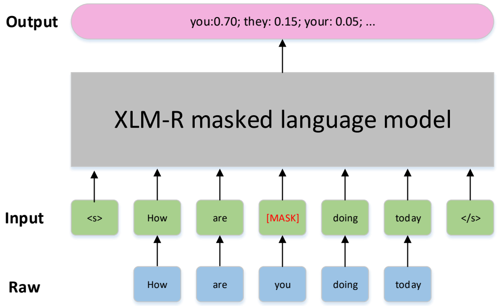

Output
you:0.70; they: 0.15; your: 0.05; ...
XLM-R masked language model
Input
<s>
How
are
[MASK]
doing
today
</s>
Raw
How
are
you
doing
today

Figure 3: An example of HWTSC-TLM metric calculation for a given sentence

For a given sentence $\boldsymbol{s} = (w_1, \ldots, w_m)$ with m tokens, the score is defined as:

$$SEG\_LM(\boldsymbol{s}) = \frac{1}{m}\sum_{i=1}^{m} \log \frac{1}{P(w_i|\boldsymbol{s} - w_i)}, \quad (1)$$

where $P(w_i|\boldsymbol{s} - w_i)$ the probability of w_i predicted by the masked language model when w_i is replaced by `[MASK]`, as shown in Figure 3. And this score is used for segment-level MT evaluation.

For system-level evaluation where a set of system translation sentences S is provided, the score is defined as:

$$SYS\_LM(S) = \frac{1}{|S|}\sum_{\boldsymbol{s}\in S} SEG\_LM(\boldsymbol{s}), \quad (2)$$

which is the mean value of SEG_LM scores on each sentence in S.

2.2.3 CrossQE

CrossQE showed as figure 4 has used pre-trained Cross-lingual XLM-Roberta large(Lample and Conneau, 2019; Conneau et al., 2019) as predictor instead of RNN-based model in the two-stage Predictor-Estimator architecture (Kim et al., 2017), and uses regressor as quality estimator, and multitasks are trained at the same time. The Cross-lingual XLM-Roberta large model is pre-trained from large-scale parallel corpora which source and target tokens are concatenated by MLM task. Shuffling those tokens and predicting those tokens' index by the pre-trained model as a additional pre-training task can improve CrossQE's effect. CrossQE is build on the COMET architecture[1] by exploring adapter layers (Houlsby et al., 2019) for quality estimation to eliminate the overfitting problem while instead of fine-tuning the whole base pre-trained model for different NLP tasks (He et al., 2021). At training step, the Mean Teacher loss(Baek et al., 2021) is added to improve model's over-fitting problem. Data augmentation method based on Monte Carlo (MC) dropout (Gal and Ghahramani, 2016) is added to enhance the performance in sentence quality score prediction.

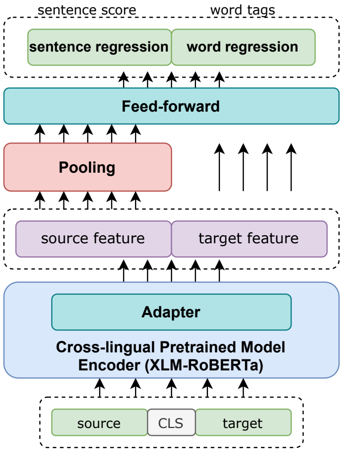


Figure 4: Cross QE architecture

2.2.4 KG-BERTScore

KG-BERTScore metric proposed by Wu et al. (2022), incorporates multilingual knowledge graph into BERTScore for reference-free MT evaluation. The evaluation process in WMT22 metrics shared task is shown in Algorithm 1:

[1] https://github.com/Unbabel/COMET

Firstly, we employ a reference-free BERTScore metric to calculate F_{BERT} score of each MT sentence. For the WMT22 metrics shared task, we use HWTSC-Teacher-Sim metric to calculate F_{BERT} so that the score is more relevant to the MQM.

Secondly, we utilize model (NER) named entity recognition to identify named entities in the sentences, and retrieve the corresponding entity IDs in multilingual knowledge graph. We then calculate F_{KG} scores based on entity matching degree. Since the same named entities in different languages share the same entity ID in multilingual knowledge graph, we can check whether they can be matched by entity IDs. For the WMT22 metrics shared task, the NER model we use is spacy[2], and the multilingual knowledge graph we use is Google Knowledge Graph Search API[3].

Finally, we combine to obtain a segment-level $F_{KG-BERT}$ score, and the $F_{KG-BERT}$ score of all MT sentences are averaged to obtain a system-level score. For the WMT 2022 metrics shared task, we set α to 0.5, and if there is no entity in the source, F_{KG} score is 1.

In addition, due to limited access to the Google Knowledge Graph Search API, we only use KG-BERTScore metric to score the three language directions zh-en, en-ru, and en-de on the WMT22 metrics shared task. The scores for other language directions in our submissions are simply populated with the F_{BERT} score based on the paraphrase-multilingual-mpnet-base-v2 model[4].

Algorithm 1: KG-BERTScore evaluation process

```
Input  : all source sentences $s_k \in S$ and
         machine translations $t_k \in T$ of
         $n$ sentence pairs
Output : a system-level score $F$
for each sentence pair $\{s_k, t_k\} \in \{S, T\}$ do
    // $x_i$, $x_j$, $\hat{x}_i$, $\hat{x}_j$ is the word embedding.
    $R_k = \frac{1}{|s_k|} \sum_{x_i \in s_k} \max_{\hat{x}_j \in t_k} x_i^T \hat{x}_j$
    $P_k = \frac{1}{|t_k|} \sum_{\hat{x}_i \in t_k} \max_{x_j \in s_k} \hat{x}_i^T x_j$
    $F_{BERT_k} = 2\frac{P_k \cdot R_k}{P_k + R_k}$
    // $entities(s_k)$, $entities(t_k)$ is the number of entities.
    if $entities(s_k) \neq 0$ then
        $F_{KG_k} = \frac{matches(entities(s_k), entities(t_k))}{entities(s_k)}$
    else
        $F_{KG_k} = 1$
    end
    // $\alpha$ is an adjustable hyperparameter.
    $F_{KG-BERT_k} = \alpha \cdot F_{KG_k} + (1 - \alpha) \cdot F_{BERT_k}$
end
$F = \frac{\sum_{k=1}^{n} F_{KG-BERT_k}}{n}$
```

3 Experiments

3.1 Experiments of Reference-based

To verify the feasibility of EE metrics 2.0, we conduct experiments mainly on WMT 20 and WMT 21 using MQM (Lommel et al., 2014) as the ground truth. To investigate the difference between when human translations are used as a system and when they are not used, we display the results computed on two sets of systems for each language pair. We report three coefficients: Pearson's correlation r, Kendall's τ and Spearman's ρ, to validate system-level correlations with human evaluations.

Table 2 displays performance comparison between EE-BERTScore and standard BERTScore, where EE-BERTScore achieves overall higher correlations with human MQM than standard BERTScore. We experiment with EE-BERTScore under different values of w, suggesting different relative weights between easy groups and difficult groups in the computation of system-level scores. We find that each setting of w is able to improve the performance of standard BERTScore, and has their best performances on a certain dataset. For example, EE-BERTScore-0.3 and EE-BERTScore-0.5 achieve a strong result on news test of WMT 20 and WMT 21, while on WMT 21 tedtalks, best performance is achieved when w is 0.8.

Since EE metrics evaluate a system relying on not only the single system, but also other participated systems, the existence of human translations may have an impact on the performances of EE metrics. As shown in Table 2, correlations with MQM drop sharply for EE-BERTScore-$*$ when human translations are included as, which is in accordance

[2] https://spacy.io/models

[3] https://developers.google.com/knowledge-graph

[4] https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2

Metric	**En→ De (w/o Human)**			**Zh→ En (w/o Human)**			**En→ Ru (w/o Human)**			**En→ De (with Human)**			**Zh→ En (with Human)**			**En→ Ru (with Human)**		
	r	τ	ρ	r	τ	ρ	r	τ	ρ	r	τ	ρ	r	τ	ρ	r	τ	ρ
				WMT 20										**WMT 20**				
BERTScore	0.754	**0.429**	**0.536**	0.742	0.643	0.810	-	-	-	0.281	**0.067**	**-0.018**	0.550	**0.422**	**0.467**	-	-	-
EE-BERTScore-0.3	0.721	**0.429**	**0.536**	**0.896**	**0.714**	**0.833**	-	-	-	**0.297**	-0.067	-0.079	**0.582**	**0.422**	**0.467**	-	-	-
EE-BERTScore-0.5	0.736	**0.429**	**0.536**	0.827	**0.714**	**0.833**	-	-	-	0.292	0.022	-0.030	0.569	**0.422**	**0.467**	-	-	-
EE-BERTScore-0.8	**0.755**	0.333	0.464	0.654	0.571	0.690	-	-	-	0.284	**0.067**	**-0.018**	0.547	0.378	0.406	-	-	-
				WMT 21-news										**WMT 21-news**				
BERTScore	0.911	0.795	**0.945**	0.577	0.308	0.484	0.776	0.538	0.692	0.181	0.441	0.500	0.382	0.295	0.439	0.540	0.417	0.485
EE-BERTScore-0.3	0.874	**0.846**	**0.945**	**0.637**	**0.487**	**0.626**	0.621	0.451	0.622	0.182	**0.485**	0.512	**0.384**	**0.410**	**0.521**	**0.569**	0.317	0.435
EE-BERTScore-0.5	0.898	**0.846**	**0.945**	0.595	0.359	0.511	0.717	0.495	0.701	0.183	0.500	0.517	0.382	0.352	0.457	0.562	0.383	0.491
EE-BERTScore-0.8	**0.919**	0.769	0.923	0.526	0.256	0.462	**0.809**	**0.604**	**0.754**	**0.184**	0.456	**0.532**	0.380	0.276	0.429	0.548	**0.467**	**0.526**
				WMT 21-tedtalks										**WMT 21-tedtalks**				
BERTScore	0.465	0.256	0.319	0.634	0.055	0.134	0.826	0.626	0.793	0.541	0.363	0.455	-0.634	-0.086	-0.079	0.659	0.676	0.832
EE-BERTScore-0.3	**0.560**	0.333	0.473	0.321	0.055	0.125	0.687	0.451	0.626	**0.553**	0.429	0.578	-0.775	-0.086	-0.086	-0.568	0.219	0.289
EE-BERTScore-0.5	0.558	0.333	0.445	0.534	**0.077**	**0.143**	0.750	0.495	0.679	0.549	0.429	0.556	-0.719	**-0.067**	**-0.071**	-0.538	0.276	0.361
EE-BERTScore-0.8	0.495	**0.359**	**0.478**	**0.645**	**0.077**	0.134	**0.829**	**0.692**	**0.829**	0.543	**0.451**	**0.582**	**-0.617**	**-0.067**	-0.079	**0.805**	**0.714**	**0.857**

Table 2: Correlations with system-level human MQM scores on datasets of WMT 20 news, WMT 21 news and WMT 21 tedtalks. EE-BERTScore-∗ represents EE-BERTScore with different w values. **With Human** indicates evaluation on MT systems and human traslations, and **w/o Human** indicates MT systems only. Best correlations are marked in bold.

with the conclusion from (Freitag et al., 2021b) that most metrics struggle to correctly score translations that are different from MT systems. However, we still see EE-BERTScore-∗ improves the correlations with human for BERTScore in some cases (En→ De in WMT 21 datasets), while there are cases where EE-BERTScore-∗ hardly has a difference with BERTScore in terms of the correlations (Zh→ En in WMT 20 news). Overall, when human translations participate as additional outputs, EE metrics bring a less significant improvement to the standard metrics.

3.2 Experiments of Reference-free

This section introduces the experimental results of our four reference-free metrics.

3.2.1 HWTSC-Teacher-Sim

We choose paraphrase-multilingual-mpnet-base-v2[4] as the model for generating sentence embeddings. Triplets were build with source, MT, and the scores of MT - the scores of MT were normalized. The MT with a higher score is closer to the source in the vector space. With TripletEvaluator, we achieve the alignment of embeddings of source and MT in the space vector. In en-de and zh-en, we use MQM data of WMT2020 and WMT2021 as train set and test set respectively. Since en-ru only has MQMdata of WMT2021, the experimental results of en-ru are missing. COMET-QE-DA_2021-src (Rei et al., 2020) is chosen as the state-of-the-art reference-free metric for comparison. And sentBLEU and BLEU (Koehn et al., 2007) are selected as the state-of-the-art reference-based metrics.

Metrics	**en-de**	**zh-en**
sentBLEU	0.083	0.176
COMET-QE-DA_2021-src	0.244	0.305
HWTSC-Teacher-Sim	0.205	0.355

Table 3: Segment-level Kendall correlations for language pairs of WMT21 MQM data

Metrics	**en-de**	**zh-en**
BLEU	0.937	0.310
COMET-QE-DA_2021-src	0.847	0.453
HWTSC-Teacher-Sim	0.863	0.596

Table 4: System-level Pearson correlations for language pairs of WMT21 MQM data

The experimental results show that the introduction of multilingual knowledge distillation is more helpful to the system level scoring accuracy of reference-free HWTSC-Teacher-Sim.

3.2.2 HWTSC-TLM

XLM-R[5] is selected as the masked language model for our metric HWTSC-TLM. The segment-level and system-level results on the 8 from-English language pairs of WMT19 are reported in Table 5 and Table 6 respectively. YiSi-2 (Lo, 2019) and Prism-src (Thompson and Post, 2020) are chosen as the state-of-the-art unsupervised reference-free metrics for comparison, and reference-based metrics sentBLEU and BLEU (Koehn et al., 2007) are selected for reference. More experimental results of HWTSC-TLM on WMT19 could be found in (Zhang et al., 2022a).

From the results in Table 5 and Table 6, it could be seen that HWTSC-TLM is much better than

[5] https://huggingface.co/xlm-roberta-base

Metrics	**en-cs**	**en-de**	**en-fi**	**en-gu**	**en-kk**	**en-lt**	**en-ru**	**en-zh**	**Avg**
sentBLEU	0.367	0.248	0.396	0.465	0.392	0.334	0.469	0.270	0.368
YiSi-2	0.069	0.212	0.239	0.147	0.187	0.003	-0.155	0.044	0.093
Prism-src	0.470	0.402	0.555	0.215	0.507	0.499	0.486	0.287	0.428
HWTSC-TLM	0.443	0.343	0.492	0.328	0.301	0.471	0.457	0.297	0.392

Table 5: Segment-level metric results for from-English language pairs of WMT19: absolute Kendall's Tau correlation of segment-level metric scores with DA.

Metrics	**en-cs**	**en-de**	**en-fi**	**en-gu**	**en-kk**	**en-lt**	**en-ru**	**en-zh**	**Avg**
BLEU	0.897	0.921	0.969	0.737	0.852	0.989	0.986	0.901	0.907
YiSi-2	0.324	0.924	0.696	0.314	0.339	0.055	0.766	0.097	0.439
Prism-src	0.865	0.976	0.933	0.444	0.959	0.908	0.822	0.793	0.838
HWTSC-TLM	0.896	0.978	0.941	0.683	0.897	0.919	0.819	0.959	0.886

Table 6: System-level metric results for from-English language pairs of WMT19: absolute Pearson correlation of system-level metric scores with DA.

YiSi-2, and is very competitive with Prism-src, which is a very strong baseline in unsupervised reference-free metrics, although only system translations are used in HWTSC-TLM.

3.2.3 CrossQE

Experiments and results of CrossQE could be found in WMT 2022 QE task report (Su et al., 2022).

3.2.4 KG-BERTScore

The ninth layer of XLM-R[5] is selected for word embedding to calculate F_{BERT} scores in our metric KG-BERTScore. The segment-level and system-level results on the 7 into-English language pairs of WMT19 are reported in Table 7 and Table 8 respectively. YiSi-2 (Lo, 2019) and reference-free BERTScore are chosen as unsupervised reference-free metrics for comparison, and reference-based metrics sentBLEU and BLEU (Koehn et al., 2007) are selected for reference. The experimental results show that the introduction of multilingual knowledge graph is more helpful to the system level scoring accuracy of reference-free BERTScore.

Metrics	de-en	fi-en	gu-en	kk-en	lt-en	ru-en	zh-en	mean
sentBLEU	0.056	0.233	0.188	0.377	0.262	0.125	0.323	0.223
YiSi-2	**0.068**	0.126	-0.001	0.096	0.075	0.053	**0.253**	0.096
BERTScore	0.036	**0.234**	**0.171**	0.310	**0.211**	0.089	0.196	**0.178**
KG-BERTScore	0.039	0.191	0.165	**0.313**	0.177	**0.095**	0.213	0.170

Table 7: Segment-level metric results for into-English language pairs of WMT19: absolute Kendall's Tau correlation of segment-level metric scores with DA.

4 Conclusions

Metrics	de-en	fi-en	gu-en	kk-en	lt-en	ru-en	zh-en	mean
BLEU	0.849	0.982	0.834	0.946	0.961	0.879	0.899	0.907
YiSi-2	0.796	0.642	-0.566	-0.324	0.442	-0.339	**0.940**	0.227
BERTScore	0.785	**0.866**	-0.007	0.117	0.657	-0.372	0.728	0.396
KG-BERTScore	**0.862**	0.733	**0.764**	**0.936**	**0.688**	**0.918**	0.908	**0.830**

Table 8: System-level metric results for into-English language pairs of WMT19: absolute Pearson correlation of system-level metric scores with DA.

In this paper, we present one reference-based metric and four reference-free metrics. We apply the methods of entropy-enhance, multilingual knowledge distillation, multilingual knowledge graph, and quality evaluation in MT to WMT 2022 Metrics Shared Task. The experimental results show great effectiveness of our research direction and the superiority of our metrics.

References

Jeonghun Baek, Yusuke Matsui, and Kiyoharu Aizawa. 2021. What if we only use real datasets for scene text recognition? toward scene text recognition with fewer labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3113–3122.

Yimeng Chen, Chang Su, Yingtao Zhang, Yuxia Wang, Xiang Geng, Hao Yang, Shimin Tao, Guo Jiaxin, Wang Minghan, Min Zhang, Yujia Liu, and Shujian Huang. 2021. HW-TSC's participation at WMT 2021 quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 890–896, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021a. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej

Bojar. 2021b. Results of the wmt21 metrics shared task: Evaluating metrics with expert-based human evaluations on ted and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jia-Wei Low, Lidong Bing, and Luo Si. 2021. On the effectiveness of adapter-based tuning for pretrained language model adaptation. *arXiv preprint arXiv:2106.03164*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Hyun Kim, Hun-Young Jung, Hongseok Kwon, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator: Neural quality estimation based on target word prediction for machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 17:1–22.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

Yilun Liu, Shimin Tao, Chang Su, Min Zhang, Yanqing Zhao, and Hao Yang. 2022. Part represents whole: Improving the evaluation of machine translation system using entropy enhanced metrics. In *Findings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Arle Lommel, Hans Uszkoreit, and Aljoscha Burchardt. 2014. Multidimensional quality metrics (mqm): A framework for declaring and describing translation quality metrics. *Revista Tradumàtica: tecnologies de la traducció*, (12):455–463.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ricardo Rei, Ana C Farinha, Chrysoula Zerva, Daan van Stigt, Craig Stewart, Pedro Ramos, Taisiya Glushkova, André F. T. Martins, and Alon Lavie. 2021. Are references really needed? unbabel-IST 2021 submission for the metrics shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1030–1040, Online.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020a. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020b. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Chang Su, Miaomiao Ma, Shimin Tao, Hao Yang, Xiang Geng, Shujian Huang, Min Zhang, Jiaxin Guo, Wang Minghan, Min Zhang, et al. 2022. Hw-tsc's participation at wmt 2022 quality estimation shared task. In *Proceedings of the Senventh Conference on Machine Translation*. Submitted.

Brian Thompson and Matt Post. 2020. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121, Online. Association for Computational Linguistics.

Minghan Wang, Hao Yang, Hengchao Shang, Daimeng Wei, Jiaxin Guo, Lizhi Lei, Ying Qin, Shimin Tao, Shiliang Sun, Yimeng Chen, and Liangyou Li. 2020. HW-TSC's participation at WMT 2020 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1056–1061, Online. Association for Computational Linguistics.

Zhanglin Wu, Min Zhang, Ming Zhu, Yinglu Li, Ting Zhu, Hao Yang, Song Peng, and Ying Qin. 2022. KG-BERTScore: Incorporating Knowledge Graph into BERTScore for Reference-Free Machine Translation Evaluation. In *11th International Joint Conference on Knowledge Graphs, IJCKG2022*. To be publiushed.

Hao Yang, Shimin Tao, Minghan Wang, Min Zhang, Daimeng Wei, Shuai Zhao, Miaomiao Ma, and Ying Qin. 2022a. CCDC: A Chinese-Centric Cross Domain Contrastive Learning Framework. In *Knowledge Science, Engineering and Management*, pages 225–236, Cham. Springer International Publishing.

Hao Yang, Minghan Wang, Daimeng Wei, Hengchao Shang, Jiaxin Guo, Zongyao Li, Lizhi Lei, Ying Qin, Shimin Tao, Shiliang Sun, and Yimeng Chen. 2020. HW-TSC's participation at WMT 2020 automatic post editing shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 797–802, Online. Association for Computational Linguistics.

Hao Yang, Min Zhang, Shimin Tao, Miaomiao Ma, Ying Qin, and Chang Su. 2022b. TeacherSim: Cross-lingual machine translation evaluation with monolingual embedding as teacher. In *The 2nd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*. To be publiushed.

Hui Yu, Xiaofeng Wu, Wenbin Jiang, Qun Liu, and Shouxun Lin. 2015. Improve the evaluation of translation fluency by using entropy of matched subsegments. *CoRR*, abs/1508.02225.

Min Zhang, Xiaosong Qiao, Hao Yang, Shimin Tao, Yanqing Zhao, Yinlu Li, Chang Su, Minghan Wang, Jiaxin Guo, Yilun Liu, and Ying Qin. 2022a. Target-side language model for reference-free machine translation evaluation. In *The 18th China Conference on Machine Translation, CCMT2022*. To be publiushed.

Min Zhang, Hao Yang, Shimin Tao, Yanqing Zhao, Xiaosong Qiao, Yinlu Li, Chang Su, Minghan Wang, Jiaxin Guo, Yilun Liu, and Ying Qin. 2022b. Incorporating multilingual knowledge distillation into machine translation evaluation. In *The 16th China Conference on Knowledge Graph and Semantic Computing, CCKS2022*. To be publiushed.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Unsupervised Embedding-based Metric for MT Evaluation with Improved Human Correlation

Ananya Mukherjee and **Manish Shrivastava**
Machine Translation - Natural Language Processing Lab
Language Technologies Research Centre
International Institute of Information Technology - Hyderabad
ananya.mukherjee@research.iiit.ac.in
m.shrivastava@iiit.ac.in

Abstract

In this paper, we describe our submission to the WMT22 metrics shared task. Our metric focuses on computing contextual and syntactic equivalences along with lexical, morphological and semantic similarity. The intent is to capture fluency and context of the MT outputs along with their adequacy. Fluency is captured using syntactic similarity and context is captured using sentence similarity leveraging sentence embeddings. The final sentence translation score is the weighted combination of three similarity scores: a) Syntactic Similarity b) Lexical, Morphological and Semantic Similarity and c) Contextual Similarity. This paper outlines two improved versions of MEE i.e., MEE2 and MEE4. Additionally, we perform our experiments on language pairs of en-de, en-ru and zh-en from WMT17-19 testset and further report the correlation with human assessments. Our submission will be made available at https://github.com/AnanyaCoder/WMT22Submission.

1 Introduction

Neural Machine Translation (NMT) systems have emerged with an increased research interest in recent times and significantly enhanced the MT quality. However, the MT research community still relies mainly on antiquated metrics and no new, universally adopted standard metric has emerged. In the last few years, research in Machine Translation (MT) evaluation has made significant progress. A metrics-shared task is held annually at the WMT conference, where new evaluation metrics are proposed and those which correlates highly with human judgements are presented from the pool of newly defined metrics. Neural-based metrics largely dominated the last two years of the WMT Metrics Task (Freitag et al., 2021; Mathur et al., 2020; Ma et al., 2019). Nevertheless, n-gram based or lexical-based metrics remain popular as automatic MT evaluation metric due to their agile and light-weighted nature. Traditionally, automatic metrics for evaluating MT quality have relied on estimating the similarity between machine outputs and reference sentences in the target language. However, advanced NMT methods yield high-quality translations that might have lexical, morphological, syntactic variations and different word choices having similar meanings. Typically, the machine output diverges from monotonic lexical transfer between the source and target languages. Widely used evaluation metrics rely on basic, lexical-level features as they calculate the surface similarity between the hypothesis and reference sentences by counting the number of matching n-grams (Papineni et al., 2002; Doddington, 2002). Metrics relying on n-gram overlap cannot appropriately capture morphological, syntactic and semantic variations as they are sensitive to only lexical variations. METEOR (Denkowski and Lavie, 2014; Gupta et al., 2010; Lavie and Denkowski, 2009; Lavie and Agarwal, 2007; Banerjee and Lavie, 2005) captures semantic variations but it is highly dependent on language specific tools . Hence, there is huge requirement for a robust, understandable, easy to use automatic MT evaluation metric which captures all the linguistic features to evaluate like humans. The better evaluation metric will be highly helpful to the development of better MT systems (Liu et al., 2011).

In this paper, we present our submission to the WMT2022 metrics shared task. We evaluate the translations of English-German (en-de), English-Russian (en-ru) and Chinese-English (zh-en) language pairs. However, the proposed metric is language independent and supports 100+ languages. Here, our submission includes scores of three metrics MEE (Mukherjee et al., 2020), MEE2 and MEE4 (MEE2 and MEE4 are extended versions of MEE). We have evaluated the testsets of WMT17 (Bojar et al., 2017), WMT18 (Bojar et al., 2018) and WMT19 (Bojar et al., 2019a,b,c), for the same

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 558 - 563
December 7-8, 2022 ©2022 Association for Computational Linguistics

language pairs (en-de, en-ru and zh-en) and reported the correlation with human assessments. The empirical results conclude that MEE4 shows better agreement with humans.

2 *M*etric for *E*valuation using *E*mbeddings (MEE)

2.1 MEE

MEE (Mukherjee et al., 2020) is an automatic evaluation metric that leverages the similarity between embeddings of words in candidate and reference sentences to assess translation quality focusing mainly on adequacy. Unigrams are matched based on their surface forms, root forms and meanings which aids to capture lexical, morphological and semantic equivalence. Semantic evaluation is achieved by using pretrained fasttext embeddings (Grave et al., 2018) provided by Facebook to calculate the word similarity score between the candidate and the reference words. MEE computes evaluation score using three modules namely exact match, root match and synonym match. In each module, fmean-score is calculated using harmonic mean of precision and recall by assigning more weightage to recall. Final translation score is obtained by taking average of fmean-scores from individual modules.

2.2 MEE2, MEE4

MEE2 and MEE4 are improved versions of MEE that capture lexical, morphological, semantic, contextual and syntactic similarity. These linguistic aspects are captured in different modules and the final sentence translation score is the weighted pool of these individual modules. Unlike MEE, these metrics capture fluency and sentence semantics. **Contextual similarity** (or sentence semantics) is obtained by computing a cosine similarity between sentence embeddings of reference sentence and system output. Whereas fluency is captured by performing **Syntactic Similarity** which is computed by using a modified BLEU score. **Lexical, Morphological and Semantic**[1] **Similarity** is measured by explicit unigram matching similar to MEE.

Figure 1 illustrates the segment-level computation of final translation score of based on a reference sentence.

2.2.1 Syntactic Similarity

Our approach assesses fluency by capturing the syntactic similarity between the reference and the hypothesis using BLEU (Papineni et al., 2002) since it follows the notion that longer n-gram scores account for the fluency of the translation. However, the length with the "highest correlation with monolingual human judgements" was found to be four (BLEU-4). Our experiments adopt the concept of BLEU with a slight variation i.e., dynamic n-gram (n depends on the sentence length). Here, while evaluating a hypothesis, the order of n-gram is based on the corresponding reference sentence length.

2.2.2 Lexical, Morphological and Semantic Similarity

In our work, *lexical, morphological and semantic* equivalence score is computed in similar to MEE metric [2]. MEE (Metric for Evaluation using Embeddings) contains three modules, namely *Exact Match, Root Match, and Synonym Match* which accounts for *lexical, morphological and semantic* features of the translation (Mukherjee et al., 2020).

2.2.3 Contextual Similarity

Contextual Similarity Score is computed by measuring the distance between the hypothesis sentence embedding and reference sentence embedding. Sentence Embedding models map text/sentences to a vector space, implying that related or similar sentences lie closer to each other in this embedding space. Sentence embedding captures the intention of the sentence. Our work is based on the assumption that contextual information of a given sentence can be captured from its vector (or embedding). We determine the context equivalence of two sentences by computing a cosine similarity (Foreman, 2014) between the embeddings of reference and hypothesis. Contextual equivalence is calculated by computing cosine similarity between the sentences embedded using LaBSE by Google AI. Out of several existing Language-Agnostic models, LaBSE (Feng et al., 2020), LASER (Artetxe and Schwenk, 2018), and Indic-Bert (Kakwani et al., 2020) we prefered to use LaBSE to embed the sentences as it is a multilingual BERT embedding model trained using MLM and TLM pre-training, resulting in a model that is effective even on low-resource languages

[1] word-level semantic similarity

[2] https://github.com/AnanyaCoder/MEE_WMT2021

for which there is no data available during training. Also, it produces language-agnostic cross-lingual sentence embeddings for 109 languages.

2.3 Score Computation

The segment-level evaluation score is computed as follows. Based on number of matched unigrams in candidate and reference sentence, individual fmean scores are computed at lexical, morphological and semantic levels. These fmean scores are achieved by parameterized harmonic mean (Sasaki, 2007) of precision and recall as per Equation 3. Ulitmately, MEE score is computed by averaging the individual fmean scores of three modules.

$$precision(P) = \frac{\#matched\_unigrams}{Total\#unigrams\_in\_hypothesis} \quad (1)$$

$$recall(R) = \frac{\#matched\_unigrams}{Total\#unigrams\_in\_reference} \quad (2)$$

$$f_\beta = \frac{(\beta^2+1)PR}{\beta^2 P + R} \quad (3)$$

MEE2 and MEE4 is computed using Equation 4 where LMS score is same as the MEE score (Mukherjee et al., 2020) i.e., $\beta = 3$ in Equation 3. Syn and Cxt are Syntax Simlarity score and Contextual Similarity score of reference and translation. The parameters in Equation 4 are manually tuned for computing MEE2 and MEE4 scores[3]. For MEE2: $\alpha = 2, \gamma = 1, \delta = 1, \epsilon = 1$ and for MEE4: $\alpha = 2, \gamma = 1, \delta = 1, \epsilon = 3$

$$score = \frac{\delta * \frac{\alpha * LMS + \gamma * Syn}{\alpha + \gamma} + \epsilon * Cxt}{\delta + \epsilon} \quad (4)$$

3 Experiments and Results

3.1 Results on WMT17-19 testset

Each year, the WMT Translation shared task organisers collect human judgements in the form of Direct Assessments. Those assessments are then used in the Metrics task to measure the correlation between metrics and therefore decide which metric works best. Therefore, we evaluated a total of 9K sentences from the testset of WMT17, WMT18, WMT19 for en-ru, en-de, zh-en language pairs and computed the pearson correlation (Benesty et al., 2009) of MEE, MEE2, MEE4 with human assessments. The segment level correlation scores are mentioned in Table 1. It is clearly evident that MEE4 correlates better with humans i.e., across the different testsets and language pairs, MEE4 demonstrates higher agreement with human judgements.

3.2 WMT22 task submission

During our experiments, we tested several techniques: averaging the module scores with different weights. Based on the agreement with humans on the WMT17-19 testset (refer Table 1, we decided to report the scores of MEE, MEE2 and MEE4 for the current WMT22 metric shared task submission. Table 2 shows the WMT22 test-set details we have experimented on.

3.2.1 Segment Level Evaluation

For Segment-level task, we submitted the sentence level scores obtained by our reference based metrics MEE2 and MEE4 for en-ru, en-de and zh-en language pairs.

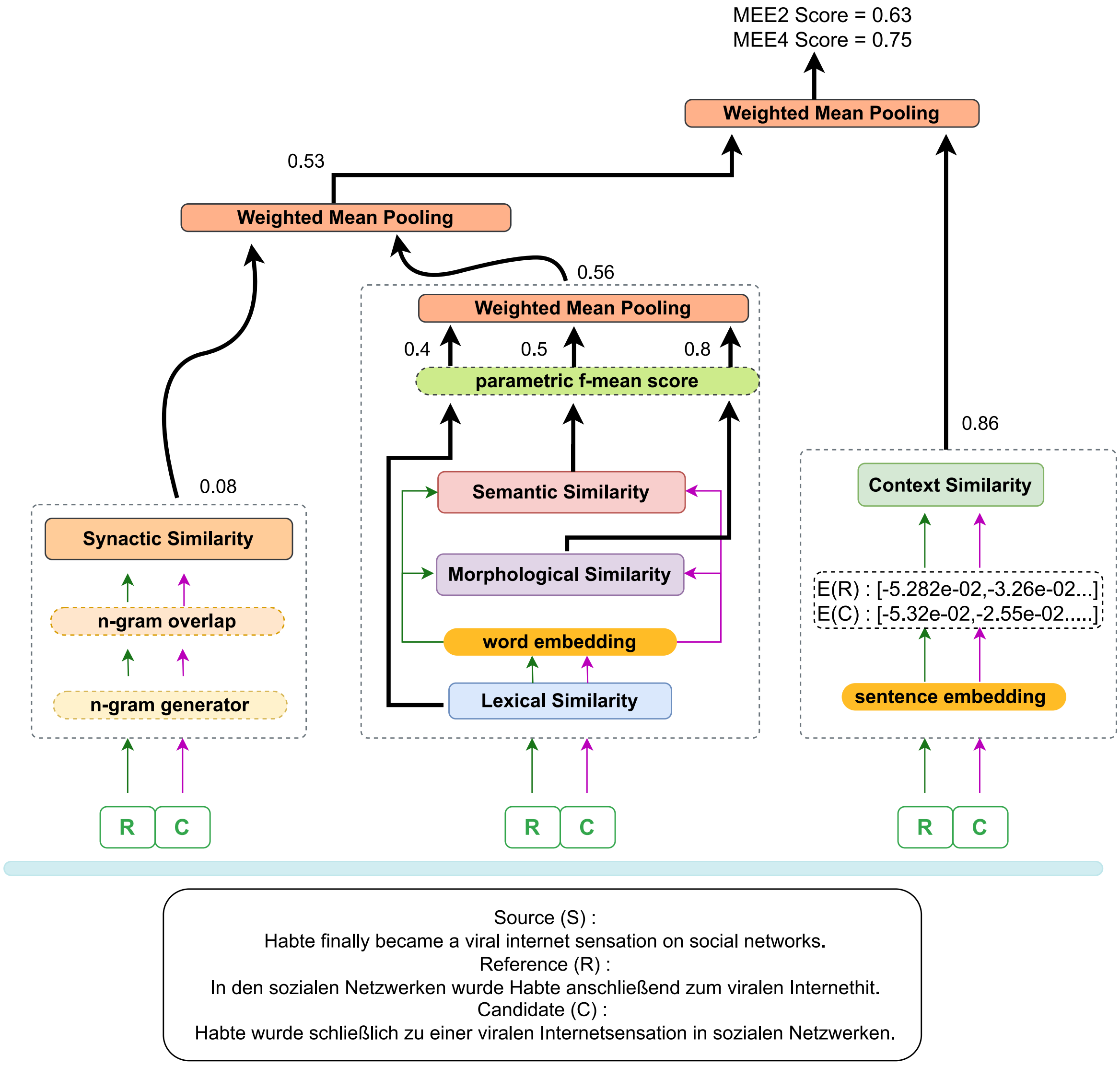
3.2.2 System Level Evaluation

For the System-level task we compute the system-level score for each system by averaging the segment-level scores obtained. We observe an equivalent approach used to compute system-level scores based on segment-level human annotations such as DA's and MQM, implying that a metric that achieves a solid segment-level correlation should also gain strong system-level performances.

4 Conclusion and Future Work

In this paper, we present our participation to the WMT22 Metrics Shared Task. Our submission includes segment-level and system-level scores for sentences of three language pairs Chinese-English (zh-en), English-Russian (en-ru) and English-German (en-de). We evaluate this year's test set using our **unsupervised, reference-based** metrics: MEE2 and MEE4. Both the metrics are extended versions of MEE with improved correlation. From the last year's findings, it was evident that MEE2 was one among the better performing metrics as it was highlighted in the top significant cluster (Freitag et al., 2021). However, this year we present MEE4 along with MEE2 and MEE4 has proved to perform better in terms of correlation with humans when evaluated on testsets of WMT17, WMT18 and WMT19. We observe that this improvement in agreement to human experts level judgements is due to assigning more weightage to context information (sentence level semantics) when compared

[3] These scores range from 0-1.



Source (S) :
Habte finally became a viral internet sensation on social networks.
Reference (R) :
In den sozialen Netzwerken wurde Habte anschließend zum viralen Internethit.
Candidate (C) :
Habte wurde schließlich zu einer viralen Internetsensation in sozialen Netzwerken.

Figure 1: Illustration of our model Architecture.

Test-set	LP	#Sentences	BLEU	MEE	MEE2	MEE4
WMT17	zh-en	1000	0.22	0.261	0.383	0.402
	en-ru	1000	0.32	0.376	0.476	0.495
	en-de	1000	0.2	0.211	0.326	0.380
WMT18	zh-en	1000	0.18	0.189	0.273	0.290
	en-ru	1000	0.32	0.335	0.404	0.414
	en-de	1000	0.42	0.476	0.549	0.563
WMT19	zh-en	1000	0.33	0.328	0.5	0.555
	en-ru	1000	0.35	0.465	0.491	0.489
	en-de	1000	0.24	0.245	0.322	0.351

Table 1: Segment Level Correlation with Human Judgements on WMT17, WMT18 and WMT19 testset.

to other linguistic aspects. In future, we plan to further experiment on optimizing weights assigned to individual linguistic modules with an aim to evaluate the translations to have better correlation with humans.

References

Mikel Artetxe and Holger Schwenk. 2018. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *CoRR*, abs/1812.10464.

Language Pair	#Sentences	#Systems
en-ru	33988	82
en-de	97002	164
zh-en	73668	192

Table 2: Data statistics of WMT22 testset for en-ru, en-de and zh-en pairs.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. *Pearson Correlation Coefficient*, pages 1–4. Springer Berlin Heidelberg, Berlin, Heidelberg.

Ondřej Bojar, Christian Buck, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, and Julia Kreutzer, editors. 2017. *Proceedings of the Second Conference on Machine Translation, Volume 1: Research Papers*. Association for Computational Linguistics, Copenhagen, Denmark.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Marco Turchi, and Karin Verspoor, editors. 2019a. *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*. Association for Computational Linguistics, Florence, Italy.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Marco Turchi, and Karin Verspoor, editors. 2019b. *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Association for Computational Linguistics, Florence, Italy.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Marco Turchi, and Karin Verspoor, editors. 2019c. *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*. Association for Computational Linguistics, Florence, Italy.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor, editors. 2018. *Proceedings of the Third Conference on Machine Translation*. Association for Computational Linguistics, Belgium, Brussels.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *the second international conference*, page 138.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT sentence embedding. *CoRR*, abs/2007.01852.

John Foreman. 2014. COSINE DISTANCE, COSINE SIMILARITY, ANGULAR COSINE DISTANCE, ANGULAR COSINE SIMILARITY.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021. Results of the wmt21 metrics shared task: Evaluating metrics with expert-based human evaluations on ted and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation*.

Ankush Gupta, Sriram Venkatapathy, and R. Sangal. 2010. Meteor-hindi : Automatic mt evaluation metric for hindi as a target language.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of EMNLP*.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

Alon Lavie and Michael J. Denkowski. 2009. The meteor metric for automatic evaluation of machine translation. *Machine Translation*, 23(2–3):105–115.

Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng. 2011. Better evaluation metrics lead to better machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 375–384, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the wmt19 metrics shared task: Segment-level and strong mt systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020. Results of the wmt20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Ananya Mukherjee, Hema Ala, Manish Shrivastava, and Dipti Misra Sharma. 2020. Mee : An automatic metric for evaluation using embeddings for machine translation. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 292–299.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318.

Yutaka Sasaki. 2007. The truth of the f-measure. *Teach Tutor Mater*.

REUSE: REference-free UnSupervised quality Estimation Metric

Ananya Mukherjee and **Manish Shrivastava**
Machine Translation - Natural Language Processing Lab
Language Technologies Research Centre
International Institute of Information Technology - Hyderabad
ananya.mukherjee@research.iiit.ac.in
m.shrivastava@iiit.ac.in

Abstract

This paper describes our submission to the WMT2022 shared metrics task. Our unsupervised metric estimates the translation quality at chunk-level and sentence-level. Source and target sentence chunks are retrieved by using a multi-lingual chunker. Chunk-level similarity is computed by leveraging BERT contextual word embeddings and sentence similarity scores are calculated by leveraging sentence embeddings of Language-Agnostic BERT models. The final quality estimation score is obtained by mean pooling the chunk-level and sentence-level similarity scores. This paper outlines our experiments and also reports the correlation with human judgements for en-de, en-ru and zh-en language pairs of WMT17, WMT18 and WMT19 testsets. Our submission will be made available at https://github.com/AnanyaCoder/WMT22Submission_REUSE

1 Introduction

Quality Estimation (QE) is an essential component of the machine translation workflow as it assesses the quality of the translated output without conferring reference translations (Specia et al., 2009; Blatz et al., 2004). High quality reference translations are often hard to find, QE helps to evaluate the translation quality based on the source sentences. Recently QE has emerged as an alternative evaluation approach for NMT systems (Specia et al., 2018). Recently, many researchers have been working on QE, as a part of Quality Estimation Shared Task, several QE systems (Zerva et al., 2021; Lim et al., 2021; Chowdhury et al., 2021; Geigle et al., 2021) were evaluated in WMT conference (Barrault et al., 2021). However, most of the quality estimation systems are supervised i.e., the model regresses on the human judgements. Often, human assessments are not available and it is very difficult to procure high quality human judgements. This motivated our research to emerge with an *Unsupervised Quality Estimation System*. Also, QE is usually performed at different granularity (e.g., word, sentence, document) (Kepler et al., 2019); in this work, we focus on the chunk-level and sentence-level similarity. The final QE score of the target sentence is obtained by mean pooling the chunk similarity scores and sentence similarity scores. Overall, our main contribution is as follows:

- We propose a concept of chunk level similarity i.e., matching the source and target chunks by leveraging multilingual BERT embeddings.
- We release a multilingual chunking model which returns meaningful word group boundaries.
- We present our unsupervised reference free QE metric (REUSE) that estimates the quality of translation by doing a chunk-level and sentence-level comparison with the source.

1.1 Motivation to use chunks

Usually, the words in translated output might not always follow the word sequence of the source text. However, it is observed that few word-groups often occur together irrespective of the order in source.

Figure 1 illustrates two example pairs: English-German (en-de) pair and English-Hindi (en-hi) pair. In the first example pair, the words sequence is not highly altered as English and German belong to the same language family (West Germanic), whereas in en-hi pair we can see a drastic change in the word order as Hindi belongs to a different language family (Indo-Aryan). However, we can observe that few word groups (here we refer as chunk) always occur *together* in both source and target. This phenomenon has motivated our research in the direction of chunk level assessment.

2 REUSE

We propose REUSE, a **RE**ference-free **U**n**S**upervised quality **E**stimation Metric that

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 564 - 568
December 7-8, 2022 ©2022 Association for Computational Linguistics

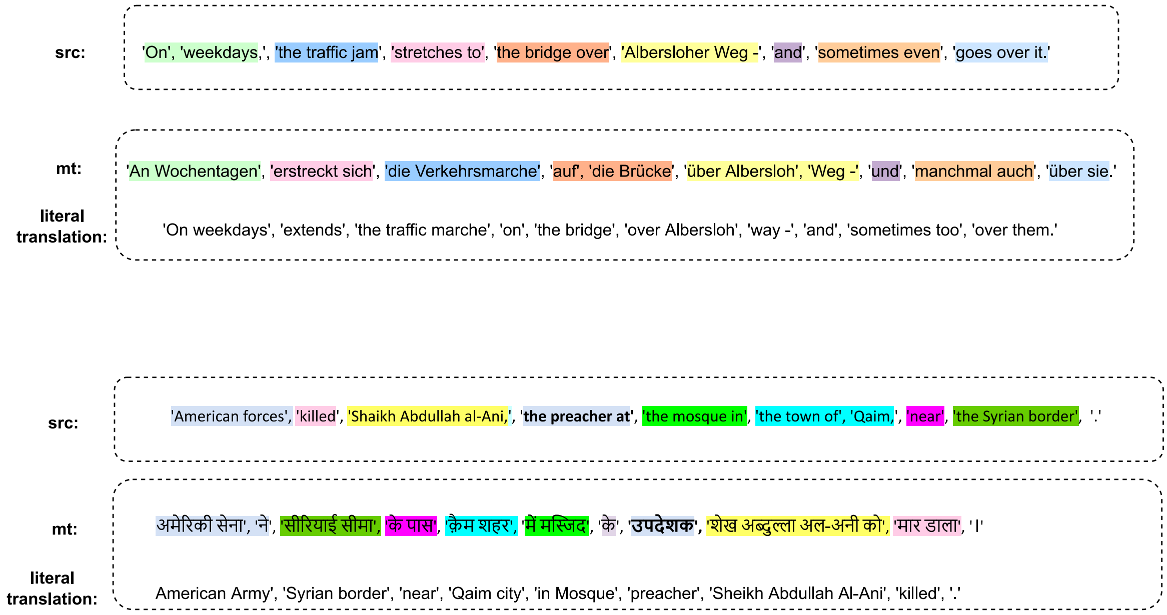

src:
'On', 'weekdays,', 'the traffic jam', 'stretches to', 'the bridge over', 'Albersloher Weg -', 'and', 'sometimes even', 'goes over it.'
mt:
'An Wochentagen', 'erstreckt sich', 'die Verkehrsmarche', 'auf', 'die Brücke', 'über Albersloh', 'Weg -', 'und', 'manchmal auch', 'über sie.'
literal translation:
'On weekdays', 'extends', 'the traffic marche', 'on', 'the bridge', 'over Albersloh', 'way -', 'and', 'sometimes too', 'over them.'
src:
'American forces', 'killed', 'Shaikh Abdullah al-Ani,', 'the preacher at', 'the mosque in', 'the town of', 'Qaim,', 'near', 'the Syrian border', '.'
mt:
अमेरिकी सेना', 'ने', 'सीरियाई सीमा', 'के पास', 'क़ैम शहर', 'में मस्जिद', 'के', 'उपदेशक', 'शेख अब्दुल्ला अल-अनी को', 'मार डाला', '।'
literal translation:
American Army', 'Syrian border', 'near', 'Qaim city', 'in Mosque', 'preacher', 'Sheikh Abdullah Al-Ani', 'killed', '.'

Figure 1: Illustration of chunk similarity for two example sentences (en-de & en-hi).

evaluates a machine translated output based on the corresponding source sentence regardless of the reference. Figure 2 depicts the high-level architecture of our model. The chunks of source and hypothesis are acquired from the multilingual chunking model. Further chunk-wise subword contextual BERT embeddings are mean-pooled to obtain the chunk-level embeddings. Meanwhile, LaBSE model (Feng et al., 2020) is used for the sentence-level embeddings. Using these embeddings, we compute chunk-level similarity and sentence-level similarity, finally combine them by averaging chunk- and sentence-level similarity scores[1]. We discuss the working details of our system in the following sections.

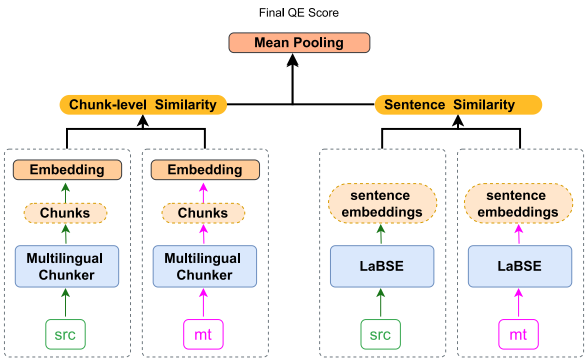


Figure 2: High-level architecture of REUSE model.

2.1 Chunk-level Similarity

We measure the number of matches between source chunks and hypothesis chunks. These matches are obtained by computing a cosine similarity (Foreman, 2014) of the individual chunk embeddings (refer 2.1.2) of source and translation sentence. An all-pair comparison is done to determine the best chunk match. Based on these matches, we compute precision and recall i.e, precision is *count of matches / length of hypothesis* and recall is *count of matches / length of source*. Ultimately, the chunk-level similarity score is calculated as the parameterized harmonic mean (Sasaki, 2007) of precision and recall, assigning more weightage to recall ($\beta = 3$).

2.1.1 Multilingual Chunker

The fundamental innovation in recent neural models lie in learning the contextualized representations by pre-training a language modeling task. Multilingual BERT is one such transformer-based masked language model that is pre-trained on monolingual Wikipedia corpora of 104 languages with a shared word-piece vocabulary. Training the pre-trained mBERT model for a supervised downstream task (finetuning) has dominated performance across a broad spectrum of NLP tasks (Devlin et al., 2018). We leverage this finetuning capability of BERT so as to create a *Multilingual Chunker* model that inputs a sentence and returns a set of divided chunks (word-groups).

[1] REUSE score ranges between 0-1.

We use **BertForTokenClassification** which has BERT (Bidirectional Encoder Representations from Transformers) as its base architecture, with a token classification head on top, allowing it to make predictions at the token level, rather than the sequence level. We use this BertForTokenClassification model and load it with the pretrained weights of "bert-base-multilingual-cased"[2]. We train the token classification head, together with the pretrained weights, using our labelled dataset (chunk annotated data). We employ Cross Entropy as the loss function and Adam optimizer (Kingma and Ba, 2014) with a learning rate of 1e-05.

2.1.2 Chunk Embeddings

Currently, we have word embedding models and sentence embedding models, but there is no specific chunk-level embedding models. Therefore, we embed the chunks leveraging the BERT embeddings by loading the weights of "distiluse-base-multilingual-cased"[3]. For a given sentence, this model return embeddings at a subword-level. To obtain the desired *chunk embeddings*, we perform a chunk to subword mapping and mean-pool the subword embeddings belonging to each chunk.

2.2 Sentence Similarity

To compute similarity at the sentence level, we find the cosine similarity (Foreman, 2014) of source sentence embedding and translation sentence embedding. We use LaBSE (Language Agnostic BERT Sentence Embedding) model to obtain the sentence embeddings. LaBSE model (Feng et al., 2020) is built on BERT architecture and trained on filtered and processed monolingual (for dictionaries) and bilingual training data. The resulting sentence embeddings achieve excellent performance on measures of sentence embedding quality, such as the semantic textual similarity (STS) benchmark and sentence embedding-based transfer learning (Feng et al., 2020).

3 Experiments and Results

3.1 Results on WMT17-19 testset

Each year, the WMT Translation shared task organisers collect human judgements in the form of Direct Assessments. Those assessments are then used in the Metrics task to measure the correlation between metrics and therefore decide which metric works best. Therefore, we estimated the translation quality of about 9K translations from the testset of WMT17 (Bojar et al., 2017), WMT18 (Bojar et al., 2018), WMT19 (Bojar et al., 2019a,b,c) for en-ru, en-de, zh-en language pairs and computed the pearson correlation (Benesty et al., 2009) of human judgements with Chunk-level Similarity scores, Sentence-level Similarity scores and their combination (REUSE). The segment level correlation scores are mentioned in Table 2. It is clearly evident from the correlations that the ensemble of Chunk Similarity model and Sentence Similarity model outperforms the individual models.

3.2 WMT22 QE-as-a-metric task submission

Table 1 shows the WMT22 QE-as-a-metric task test-set details for the language pairs we have experimented on.

Language Pair	#Sentences	#Systems
en-ru	36723	88
en-de	82356	91
zh-en	41127	103

Table 1: Data statistics of WMT22 QE-as-a-metric task testset for en-ru, en-de and zh-en pairs.

3.2.1 Segment Level Evaluation

For Segment-level task, we submitted the sentence level scores obtained by our reference free quality estimation metric (REUSE) for en-ru, en-de and zh-en language pairs.

3.2.2 System Level Evaluation

We compute the system-level score for each system by averaging the segment-level scores obtained. A similar method is also used to compute system-level scores based on segment-level human annotations such as DA's and MQM, implying that a metric with a high segment-level correlation should also demonstrate high system-level correlation.

4 Conclusion

In this paper, we describe our submission to the WMT22 Metrics Shared Task (QE-as-a-metric). Our submission includes segment-level and system-level quality estimation scores for sentences of three language pairs Chinese-English (zh-en), English-Russian (en-ru) and English-German (en-de). We evaluate this year's test set using our **unsupervised, reference-free** metric - REUSE, that

[2] https://huggingface.co/bert-base-multilingual-uncased

[3] https://huggingface.co/distiluse-base-multilingual-cased

WMT test-set	Language Pair	Chunk Similarity using chunker	Sentence Similarity using LaBSE	REUSE (chunk + sentence)
wmt17	zh-en	0.269	0.242	0.316
	en-ru	0.308	0.223	0.337
	en-de	0.280	0.167	0.278
wmt18	zh-en	0.135	0.2	0.210
	en-ru	0.145	0.2	0.213
	en-de	0.306	0.107	0.273
wmt19	zh-en	0.225	0.279	0.3
	en-ru	-0.112	0.144	-0.003
	en-de	0.254	0.131	0.251

Table 2: Correlation with Human Judgements on WMT17, WMT18 and WMT19 testset.

provides a quality estimation score by evaluating a hypothesis against the source sentence. REUSE estimates the translation quality by combining chunk-level similarity score and sentence-level similarity score, leveraging multilingual BERT embeddings. We performed our experiments on testsets of WMT17, WMT18, WMT19 and it has been empericaly observed that the combination of *chunk- and sentence-level* similarity scores performed better in terms of agreement with human assessments.

Potential research directions definitely include improving the multilingual chunking model. As part of future work, we aim to further experiment and emerge with such effortless efficient unsupervised approach to estimate the translation quality and exhibit higher agreement with humans.

References

Loic Barrault, Ondrej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussa, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Tom Kocmi, Andre Martins, Makoto Morishita, and Christof Monz, editors. 2021. *Proceedings of the Sixth Conference on Machine Translation*. Association for Computational Linguistics, Online.

Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. *Pearson Correlation Coefficient*, pages 1–4. Springer Berlin Heidelberg, Berlin, Heidelberg.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 315–321, Geneva, Switzerland. COLING.

Ondřej Bojar, Christian Buck, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, and Julia Kreutzer, editors. 2017. *Proceedings of the Second Conference on Machine Translation, Volume 1: Research Papers*. Association for Computational Linguistics, Copenhagen, Denmark.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Marco Turchi, and Karin Verspoor, editors. 2019a. *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*. Association for Computational Linguistics, Florence, Italy.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Marco Turchi, and Karin Verspoor, editors. 2019b. *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Association for Computational Linguistics, Florence, Italy.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Marco Turchi, and Karin Verspoor, editors. 2019c. *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*. Association for Computational Linguistics, Florence, Italy.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor, editors. 2018. *Proceedings of the Third Conference on Machine*

Translation. Association for Computational Linguistics, Belgium, Brussels.

Shaika Chowdhury, Naouel Baili, and Brian Vannah. 2021. Ensemble fine-tuned mbert for translation quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 897–903, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT sentence embedding. *CoRR*, abs/2007.01852.

John Foreman. 2014. COSINE DISTANCE, COSINE SIMILARITY, ANGULAR COSINE DISTANCE, ANGULAR COSINE SIMILARITY.

Gregor Geigle, Jonas Stadtmüller, Wei Zhao, Jonas Pfeiffer, and Steffen Eger. 2021. Tuda at wmt21: Sentence-level direct assessment with adapters. In *Proceedings of the Sixth Conference on Machine Translation*, pages 911–919, Online. Association for Computational Linguistics.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, António Góis, M. Amin Farajian, António V. Lopes, and André F. T. Martins. 2019. Unbabel's participation in the WMT19 translation quality estimation shared task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 78–84, Florence, Italy. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Seunghyun Lim, Hantae Kim, and Hyunjoong Kim. 2021. Papago's submission for the wmt21 quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 935–940, Online. Association for Computational Linguistics.

Yutaka Sasaki. 2007. The truth of the f-measure. *Teach Tutor Mater*.

Lucia Specia, Carolina Scarton, and Gustavo Paetzold. 2018. Quality estimation for machine translation. *Synthesis Lectures on Human Language Technologies*, 11:1–162.

Lucia Specia, Marco Turchi, Nicola Cancedda, Nello Cristianini, and Marc Dymetman. 2009. Estimating the sentence-level quality of machine translation systems. In *Proceedings of the 13th Annual conference of the European Association for Machine Translation*, Barcelona, Spain. European Association for Machine Translation.

Chrysoula Zerva, Daan van Stigt, Ricardo Rei, Ana C Farinha, Pedro Ramos, José G. C. de Souza, Taisiya Glushkova, miguel vera, Fabio Kepler, and André F. T. Martins. 2021. Ist-unbabel 2021 submission for the quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 961–972, Online. Association for Computational Linguistics.

MaTESe: Machine Translation Evaluation as a Sequence Tagging Problem

Stefano Perrella[1], **Lorenzo Proietti**[1], **Alessandro Scirè**[1,2]
Niccolò Campolungo[1] and **Roberto Navigli**[1]
[1]Sapienza NLP Group, Sapienza University of Rome
[2]Babelscape, Italy
{stefano.perrella, l.proietti, alessandro.scire}@uniroma1.it
campolungo@di.uniroma1.it navigli@diag.uniroma1.it

Abstract

Starting from last year, WMT human evaluation has been performed within the Multidimensional Quality Metrics (MQM) framework, where human annotators are asked to identify error spans in translations, alongside an error category and a severity. In this paper, we describe our submission to the WMT 2022 Metrics Shared Task, where we propose using the same paradigm for automatic evaluation: we present the MaTESe metrics, which reframe machine translation evaluation as a sequence tagging problem. Our submission also includes a reference-free metric, denominated MaTESe-QE. Despite the paucity of the openly available MQM data, our metrics obtain promising results, showing high levels of correlation with human judgements, while also enabling an evaluation that is interpretable. Moreover, MaTESe-QE can also be employed in settings where it is infeasible to curate reference translations manually.

1 Introduction and Related Work

For many years, Machine Translation (MT) has mainly been evaluated using untrained evaluation techniques, such as BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) and chrF (Popović, 2015), which rely heavily on lexical-level matching of either token, or character, n-grams. Unfortunately, these metrics present two major drawbacks: i) it is not possible to carry out the evaluation without manually-curated references and, most importantly, ii) the evaluation is too dependent on the surface form of the translation, and its reference. More recently, attempts have been made to address these problems using machine-learned metrics, which have shown better correlations with human judgements (Mathur et al., 2020). More specifically, last year's WMT Metrics Shared Task saw C-SPEC$_{\text{PN}}$ (Takahashi et al., 2021), BLEURT-20[1] and COMET-MQM_2021 (Rei et al., 2021) emerge as distinctly better than the other participants (Freitag et al., 2021b). These metrics consist of regression models trained to mimic human annotators by directly assigning quality scalar scores to candidate translations. In detail, COMET-MQM_2021 is based on the Estimator architecture introduced by Rei et al. (2020), where features extracted from the embeddings of the source sentence, candidate translation, and reference translation are passed to a feed-forward regressor; C-SPEC first concatenates the embeddings derived from paired inputs of candidate-source and candidate-reference, and then passes the resulting vector to a multi-layer perceptron; BLEURT, instead, feeds the candidate translation and its reference to Rebalanced mBERT (Chung et al., 2021), and regresses on the representation provided by the [CLS] token. Moreover, BLEURT and C-SPEC add automatically-generated negative pairs to the standard training data: BLEURT applies random token perturbations, while C-SPEC uses Word Attribute Transfer to replace words in the translations. Although undoubtedly effective, regression metrics have the major drawback of not being interpretable, meaning that users are not able to gauge the quality of assessments that are returned, which is of paramount importance for an evaluation metric.

Recently, Freitag et al. (2021a) have proposed a shift in the standard practices for human machine translation evaluation, employing the Multidimensional Quality Metrics framework (Lommel et al., 2014, MQM), and moving away from Direct Assessments (Graham et al., 2013, DA), which were computed via requiring (even non-expert) annotators to assign a scalar value to a candidate translation, given a reference. Furthermore, Freitag et al. (2021a) pointed out the limitations of non-professional Direct Assessments, also show-

[1] BLEURT-20 is the retrained version of the previous year's BLEURT submission (Sellam et al., 2020).

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 569 - 577
December 7-8, 2022 ©2022 Association for Computational Linguistics

ing their unreliability compared to MQM. Indeed, differently from Direct Assessments, annotators who follow the MQM guidelines look at the source sentence rather than the reference, and are expected to tag the spans of the candidate translations that contain errors,[2] together with their error category (e.g., `Fluency/Grammar`, `Fluency/Punctuation` or `Style/Awkward`) and severity (e.g., `Major` or `Minor`), which, combined, determine the score associated with the error span. Finally, a scalar quality score for the entire sentence is derived from the various annotated spans.

In this work, we introduce the MATESE and MATESE-QE metrics, reframing the evaluation of machine-translated text as a sequence tagging problem based on the MQM framework, in an attempt to develop metrics that are interpretable, while also displaying high levels of correlation with human judgements.

2 MATESE Metrics

Inspired by the novel MQM evaluation framework, our work aims at employing a similar paradigm for automatic evaluation. We propose the MATESE metrics which, given a candidate translation and its reference (or source, for MATESE-QE), assign a label to each token of the candidate. These labels identify error spans, together with their severity, chosen among `Major` and `Minor`. Finally, in order to associate a score with the entire tagged sentence, we follow a weighting scheme similar to the one presented by Freitag et al. (2021a) for MQM-based human evaluation: we assign a score to an entire error span based on its severity, i.e., -5 and -1 for `Major` and `Minor`, respectively. The score assigned to a translation is the sum of the scores assigned to its error spans, with a minimum total score of -25. Following Freitag et al. (2021a), we compute a corpus-level score by averaging the scores of the sentences in the corpus. Although human MQM annotators are asked to report a maximum of 5 errors per translation,[3] we decided to let our metrics detect as many errors as they can find; nevertheless, in order to keep our scores in the same range as those computed on gold MQM annotations, we set a minimum score of -25, which is equal to the sum of 5 `Major` errors. Figure 1 shows an example of the annotations returned by our metrics.

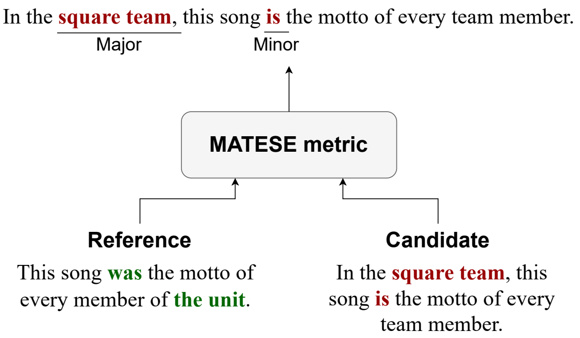

In the square team, this song is the motto of every team member.
Major
Minor
MATESE metric
Reference
This song was the motto of every member of the unit.
Candidate
In the square team, this song is the motto of every team member.

Figure 1: Example of the annotation returned by the MATESE metrics, given a candidate translation and its reference. The final score of the translation is -6, that is the sum of -5 and -1, assigned to the `Major` and `Minor` errors, respectively.

2.1 Data pre-processing

According to the MQM guidelines, mistranslated spans are tagged with an error category and a severity. To reduce the granularity of the annotations, we apply some transformations to the original data, which we report below:

1. We discard annotations of the `Non-translation` category, since they are weighted -25 by Freitag et al. (2021a), and would have required a special treatment, but are too scarce ($< 0.1\%$ of the whole data) for the model to learn how to assign them;
2. We discard annotations referring to either `Accuracy/Omission` or `Source error` categories, since in these cases the annotation might be in the source sentence, while our models are trained to tag the candidate translation only;
3. We discard annotations of errors with `Neutral` severity, since they are highly subjective and do not participate in the computation of the final quality score (Freitag et al., 2021a);
4. We replace `Critical` severity labels with `Major`, in order to make the English→Russian dataset conform to the rest of the data;
5. We discard all the MQM error categories, leaving only information about error severity. While we believe error categorization to

[2] In a few cases, the source sentence might also be annotated. An example of this is with omission errors, where annotators report the spans of the source sentence which are missing from the candidate translation.

[3] This holds only for the MQM guidelines released by Freitag et al. (2021a).

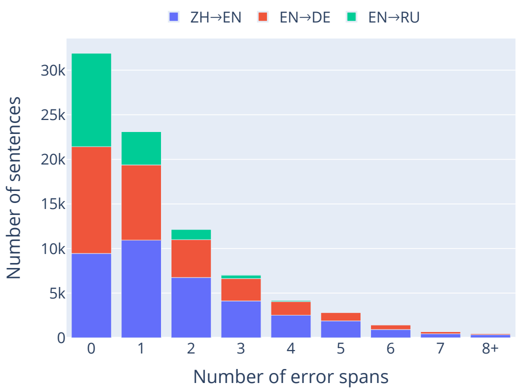


Figure 2: Distribution of the number of error spans over sentences in both training and test data of WMT 2021 Metrics Shared Task, after the pre-processing we described in Section 2.1.

be of great importance, we decided to remove it because of the limited availability of training data and to avoid making the classification problem too sparse.

Furthermore, in the MQM data released by Freitag et al. (2021a), every sentence has been annotated 3 times, each one by a different rater. In order to yield a single sample per sentence and maximize the number of annotations, we merge the annotations of the different raters into a single annotated sentence;[4] in the case when there is even only a partial overlap between two annotated spans, we discard the one associated with the `Minor` error in favor of the `Major`, or pick one or the other randomly if they have the same severity. We decided to keep `Majors` over `Minors` because Freitag et al. (2021a) obtained almost the same ranking of MT systems when considering only `Major` errors, compared to the full MQM score.

2.2 Hypothesis and Target Span Hit metrics

Typically, MT evaluation metrics' quality is assessed through their correlations with human judgements. Nevertheless, our novel formulation of MT evaluation as a sequence tagging problem allows us to estimate the quality of our metrics also via the produced error spans. Specifically, we are interested in determining how well our metrics are able to flag, even partially, a true error span, regardless of its severity or length. However, existing span-level metrics, such as Span Precision, Span Recall and Span F1, focus on exact overlaps between predicted spans and target ones. Moreover, correlations with MQM scores paint only a partial picture, since the final score assigned to a translation depends only on the number of error spans (with their severity), but not on their position in the sentence. For instance, if a system flagged a span as a `Major` error, but the target annotation had a different span tagged as `Major`, the MQM scores would be identical despite the tagging error.

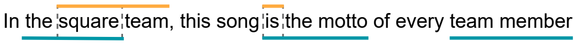

In the square team, this song is the motto of every team member

Figure 3: An example of evaluation with the Hypothesis and Target Span Hit metrics. The **turquoise line —** (below) and **amber line —** (above) represent the hypothesis and target annotation, respectively. HSH $= 2/3$ (2 out of 3 spans are *hit*), TSH $= 2/2$ (2 out of 2 spans are *hit*).

To address these issues, we introduce the Hypothesis Span Hit (HSH) and Target Span Hit (TSH) metrics: HSH represents the percentage of predicted error spans that are also, at least partially, true; instead, TSH represents the percentage of true error spans that the metric has predicted, even partially. An example of their assessments is given in Figure 3.

Formal definition Let us consider a candidate translation $\boldsymbol{c}$ as a sequence of tokens $(c_1, c_2, \dots, c_n)$; moreover, let us define an error span s as a set of contiguous tokens in $\boldsymbol{c}$, e.g., $\{c_1, c_2, c_3\}$, and an error annotation A as a set of disjoint error spans, i.e., that satisfies $\bigcap_{s' \in A} s' = \emptyset$. Furthermore, we define the Span Hit Indicator as

$$\mathrm{SHI}(s, A) = \mathbb{I}(s \cap \sigma(A) \neq \emptyset)$$

where $\sigma(A) = \bigcup_{s' \in A} s'$, i.e., the set of all tokens in annotation A. In simpler terms, $\mathrm{SHI}(s, A)$ is 1 if at least one of the tokens in s belongs to the set of all tokens of the error spans in A.

Finally, let us take two error annotations: A_h represents the hypothesis spans produced by a model, while A_t represents the target spans that $\boldsymbol{c}$ was originally annotated with. We define the Hypothesis Span Hit and Target Span Hit metrics as follows:

$$\mathrm{HSH}(A_h, A_t) = \frac{\sum_{s_h \in A_h} \mathrm{SHI}(s_h, A_t)}{|A_h|}$$

$$\mathrm{TSH}(A_t, A_h) = \frac{\sum_{s_t \in A_t} \mathrm{SHI}(s_t, A_h)}{|A_t|}$$

[4] Therefore, in our merged sentences the number of error spans per translation can be greater than 5. Figure 2 reports the distribution of error spans in our entire data.

Both metrics are defined as the average number of span hits of one error annotation with respect to the other. To compute the metrics for an entire dataset we employ micro-averaging, i.e., we concatenate all hypotheses into a single one, do the same for the targets, and then measure Span Hit metrics on the newly-created pair of hypothesis and target. We avoid averaging the single results because the number of spans varies widely across samples (Figure 2).

3 Experimental Setup

In this Section, we describe the different architectures we experiment with, the data for training and evaluation, and the metrics we use to measure performances.

3.1 Architectures

Since it is rather convenient to have a single model capable of evaluating text in multiple languages, we leverage multilingual pre-trained models like XLM-RoBERTa (Conneau et al., 2020) and mBART (Liu et al., 2020). In order to compare the performances of multilingual models with their English-only counterparts, we also experiment with RoBERTa (Liu et al., 2019).[5]

Encoder-only models XLM-RoBERTa and RoBERTa models consist of only the encoder part of the standard Transformer architecture (Vaswani et al., 2017). The input we provide to the encoder models is the concatenation of the candidate translation and its reference (or source, for MATESE-QE), separated by a </s> token. Furthermore, we add two randomly-initialized encoder layers on top of the last layer, as well as a classification head. Due to computational constraints, we keep the embedding layer frozen.

Encoder-decoder model When experimenting with mBART, we feed the reference translation (or the source, for MATESE-QE) to the encoder, and the candidate to the decoder, so as to maintain similarity with the pre-training process. We highlight that we do not use the decoder autoregressively; instead, following the standard practice for sequence classification with encoder-decoder models, we force the candidate to be processed all at once, and collect the contextualized embeddings at the last layer. On top of the decoder, we add two randomly-initialized encoder layers, and a classification head. As with the encoder-only models, due to computational constraints the embedding layer is frozen.

3.2 Training and validation data

In order to perform our experiments employing all the existing MQM data, we experiment using a 90/10 training/validation split of the concatenation of the training set (which is the MQM data released by Freitag et al. (2021a)) and the test sets of WMT 2021 Metrics Shared Task (Freitag et al., 2021b).

Moreover, to make a fair comparison between the MATESE metrics and the ones submitted to the aforementioned Shared Task, we also retrain our systems using only the above-mentioned training set, with the same split. We dub these systems MATESE^{21} and MATESE-QE^{21}.

In both settings, we use only English→German and Chinese→English data. Moreover, we point out that the split is performed on *unique source sentences*: since each source sentence is translated by multiple systems, our split avoids having translations of the same source sentence be present in both the training and validation splits.

WMT Submission Training Split For our final submission to the WMT 2022 Metrics Shared Task, we include English→Russian data to the concatenation of the training and test sets of the WMT 2021 Metrics Shared Task. We split the whole data 5 times, each time taking 90% for training and 10% for validation, and train 5 different systems (10 if we also consider MATESE-QE). In our submission, each score is the median prediction of the systems trained on the 5 different data splits.

3.3 Evaluation metrics

The MATESE metrics tag the spans of a candidate translation that contain an error. Following the BIO scheme (Ramshaw and Marcus, 1995), we assign to each token a label in $L = \{$`O`, `B-Minor`, `I-Minor`, `B-Major`, `I-Major`$\}$; a final score for the annotated sentence is then obtained as the sum of the individual spans' scores. We can evaluate the performances of our metrics according to the final scores, as well as in terms of the produced annotations: indeed, we use the scalar scores to rank translations and measure the correlations with human judgements, and we measure the tagging accuracy with respect to the gold annotations. In the latter

[5] RoBERTa can be employed only for reference-based evaluation, and with language pairs that have English as target language: in our case, this is only Chinese→English.

	O	B-Minor	I-Minor	B-Major	I-Major
EN→DE	818,945	32,667	37,897	8516	25,192
ZH→EN	1,053,663	33,633	48,333	33,996	76,984
EN→RU	343,449	614	1015	7271	3189
ALL	2,216,057	66,914	87,245	49,783	105,365

Table 1: Distribution of the token-level gold annotations in the concatenation of the training and test sets of WMT 2021 Metrics Shared Task, after the pre-processing we described in Section 2.1.

	P	R	F1	HSH	TSH
XLM-R$^{\text{LARGE}}$	47.38	**38.40**	**41.72**	57.73	**46.08**
XLM-R$^{\text{BASE}}$	46.64	34.12	37.93	**58.01**	38.70
mBART	**47.97**	31.94	36.01	55.85	32.66

Table 2: Comparison of different architectures in terms of Precision, Recall and F1-score in their macro versions; HSH and TSH are Hypothesis Span Hit and Target Span Hit metrics.

case, we rely on the standard classification metrics of Precision, Recall and F1-score, computed using TorchMetrics[6] modules. More specifically, given that our data is highly imbalanced (see Table 1), we employ macro versions of these metrics and, in particular, use the macro-F1 score to select the best checkpoint of the models on the validation set. Furthermore, in order to assess the span-level error detection capabilities of our systems, we employ the Hypothesis and Target Span Hit metrics as defined in Section 2.2.

4 Results

In this Section, we show the results obtained by our metrics. Unless explicitly specified, all experiments have been performed using reference-based systems.

4.1 Architectures comparison

We can see the results of comparing the aforementioned architectures in Table 2. The best performing architecture is XLM-R$^{\text{LARGE}}$, which attains the highest F1-score, as a consequence of achieving the best Recall. Considering the complexity of the task, and the imbalance of the data, we conjecture that the other architectures obtain high Precision and low Recall scores because they are able to predict only the errors that are easier to detect, while assigning `O`s more frequently. This is also confirmed by the TSH score which, ruling `O` labels out of the computation, exacerbates the difference between different architectures, with XLM-R$^{\text{BASE}}$ and mBART clearly failing to detect a higher number of errors of the target annotation compared to XLM-R$^{\text{LARGE}}$. An additional interesting fact that emerges from this comparison is that XLM-R architectures perform better than mBART, with XLM-R$^{\text{BASE}}$ outperforming it despite having less than half of its parameters.

4.2 Monolingual-multilingual comparison

Table 3 reports the results of training the same XLM-R model using a single language pair at a time, or both. Moreover, we test whether an English language model like RoBERTa outperforms XLM-R, when dealing with English-only data. Our results show that training on the whole data is beneficial to the task, with XLM-R$^{\text{ALL}}$ obtaining a higher Recall and Target Span Hit in both language pairs, and an F1-score that is higher, or on par with, its variants. Similarly to what happens with different architectures, we hypothesize that training on more data enables the models to detect a wider range of errors, even if the additional data is in a different language. We do not record significant differences in the results obtained by RoBERTa, compared to XLM-R$^{\text{MONO}}$ on Chinese→English data.

4.3 MaTESe-QE

A desirable feature of evaluation metrics is to function both in the presence and the absence of humanly-curated references. To achieve this, we investigate whether it is feasible to tag the errors in the candidate translation by looking at the source sentence only. Table 4 reports the results obtained by the best architecture, i.e., XLM-R$^{\text{LARGE}}$, trained on both English→German and Chinese→English, both when disposing of the reference sentence, and not.

MaTESe outperforms MaTESe-QE in terms of Recall, F1-score and Target Span Hit metrics. Clearly, the information found in the reference is easier to exploit, and the reference-based system is able to detect a much wider range of errors. At the same time, MaTESe-QE proves to be a viable alternative in the absence of manually-curated references: it displays high levels of Precision and Hypothesis Span Hit, which means that it outputs predictions that are more accurate than those of MaTESe, even if only for the range of errors that it is able to detect.

[6] https://github.com/Lightning-AI/metrics

	EN→DE					ZH→EN				
	P	R	F1	HSH	TSH	P	R	F1	HSH	TSH
XLM-R$^{\text{ALL}}$	43.03	**33.09**	**35.89**	54.35	**44.54**	47.86	**39.39**	42.63	59.90	**47.04**
XLM-R$^{\text{MONO}}$	**43.98**	30.64	33.52	**56.56**	39.67	50.85	38.51	**42.77**	63.51	44.38
RoBERTa	–	–	–	–	–	**51.49**	37.91	42.41	**64.33**	42.81

Table 3: Model performances on monolingual and multilingual settings. XLM-R$^{\text{ALL}}$ is trained and evaluated on the concatenation of English→German (EN→DE) and Chinese→English (ZH→EN) datasets, while XLM-R$^{\text{MONO}}$ stands for two different models, each one trained and evaluated on a single dataset. RoBERTa is an English language model, and therefore can deal with ZH→EN data only.

	P	R	F1	HSH	TSH
MaTESe	47.38	**38.40**	**41.72**	57.73	**46.08**
MaTESe-QE	**49.34**	34.53	38.89	**59.89**	36.84

Table 4: Comparison of our reference-based and reference-free systems, i.e., MaTESe and MaTESe-QE, respectively. The only difference between the two is that MaTESe-QE uses the source sentence in place of the reference.

4.4 Correlations with Human Judgements

Tables 5a and 5b report the correlations with human judgements that our metrics attained on `newstest2021` (in-domain) and `TED` (out-of-domain) test sets of last year's WMT Metrics Shared Task: `w/ HT` means that manually-curated references have been scored together with system outputs, while `w/o HT` means that those references have been kept out of the evaluation. Aside from our systems, i.e., MaTESe^{21} and MaTESe-QE^{21}, we also report two additional baselines: #1 WMT and #2 WMT. These are the top-1 and top-2 results reported by Freitag et al. (2021b) in the corresponding tables (Tables 23, 24, 27 and 28). Since those positions are held by different systems, we assign each submission a unique symbol and report the mapping in Appendix A.

Generally speaking, for in-domain settings, we observe that, on English→German, MaTESe^{21} and MaTESe-QE^{21} achieve correlations on par or better than the top-2 WMT 2021 submissions, while on Chinese→English the results are slightly worse. Interestingly, in out-of-domain settings, we observe a sizeable drop in correlation on both translation directions. We attribute this drop to the very limited amount of training data, which probably hinders proper generalization capabilities to out-of-domain settings. Finally, we observe that MaTESe-QE^{21} lags behind MaTESe^{21} by a relatively small margin.

	EN→DE			ZH→EN		
	w/o HT	w/ HT	TED	w/o HT	w/ HT	TED
#1 WMT	‡0.938	$^{\perp}$0.823	$^{\parallel}$**0.818**	$^{\wedge}$**0.834**	$^{\wedge}$**0.727**	$^{\vee}$**0.421**
#2 WMT	†0.937	$^{\perp}$0.822	$^{\perp}$0.802	$^{\parallel}$0.628	$^{\parallel}$0.619	$^{\top}$0.403
MaTESe^{21}	**0.946**	**0.863**	0.621	0.636	0.701	0.017
MaTESe-QE^{21}	0.910	0.806	0.584	0.502	0.600	0.056

(a) System-level Pearson correlations.

	EN→DE			ZH→EN		
	w/o HT	w/ HT	TED	w/o HT	w/ HT	TED
#1 WMT	$^{\perp}$0.267	$^{\perp}$0.256	$^{\wedge}$**0.290**	$^{\perp}$**0.402**	$^{\perp}$**0.390**	$^{\wedge}$0.248
#2 WMT	$^{\perp}$0.266	$^{\perp}$0.254	$^{\perp}$0.285	$^{\perp}$0.401	$^{\perp}$0.388	$^{\perp}$0.241
MaTESe^{21}	**0.323**	**0.310**	0.271	0.358	0.346	**0.257**
MaTESe-QE^{21}	0.288	0.277	0.210	0.343	0.332	0.196

(b) Segment-level Kendall correlations.

Table 5: System- and segment-level correlations with human judgements as measured in WMT 2021 Metrics Shared Task (Freitag et al., 2021b). MaTESe^{21} and MaTESe-QE^{21} are MaTESe metrics that have been re-trained using only the training set of the Shared Task. A legend of the other symbols is found in Appendix A.

5 Conclusions

In this paper, we described our submission to the WMT 2022 Metrics Shared Task: we presented the MaTESe metrics, a new way of automatically assessing the quality of translations, putting forward evaluation techniques that are interpretable, while at the same time displaying high levels of correlation with human judgements. Scores are in the same ballpark of the best performing metrics proposed in the WMT 2021 Metrics Shared Task. Furthermore, the MaTESe metrics can also be used in the absence of humanly-curated references, with MaTESe-QE being slightly less accurate than its reference-based counterpart, but still presenting encouraging levels of correlation with human judgements. In future work, we plan to improve the MaTESe metrics to also detect the type of errors, and not only their severity, in order to approximate even better the MQM annotation process.

Acknowledgements

The authors gratefully acknowledge the support of the ERC Consolidator Grant MOUSSE No. 726487.

The authors also acknowledge the support of the PerLIR project (Personal Linguistic resources in Information Retrieval) funded by the MIUR Progetti di ricerca di Rilevante Interesse Nazionale programme (PRIN 2017).

Limitations

Poor generalization We expect the MaTESe metrics' generalization capabilities to be hindered by the narrow range of errors that they are trained upon. Indeed, while the number of samples in the datasets is relatively large (around 80K annotated sentences), the number of unique sources is much smaller (around 6K), because the annotations are performed on the same source sentences translated by multiple MT systems. In fact, we observe a drop in performance in the out-of-domain setting, i.e., the TED dataset.

Computational requirements The MaTESe metrics require a non-negligible computational budget, especially when compared to their untrained alternatives, such as BLEU, METEOR or chrF. Given that the task we tackle is arguably challenging, and that we need semantically-rich representations of the analyzed sentences, we decided to rely upon a large Transformer encoder, which makes the evaluation computationally intensive. Unfortunately, the comparison between XLM-RoBERTa Large and its Base counterpart shows that a sizeable improvement is due to the increased size of the model.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking embedding coupling in pre-trained language models. In *International Conference on Learning Representations*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021a. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021b. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Lifeng Han, Irina Sorokina, Gleb Erofeev, and Serge Gladkoff. 2021. cushLEPOR: customising hLEPOR metric using optuna for higher agreement with human judgments or pre-trained language model LaBSE. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1014–1023, Online. Association for Computational Linguistics.

Mateusz Krubiński, Erfan Ghadery, Marie-Francine Moens, and Pavel Pecina. 2021a. Just ask! evaluating machine translation by asking and answering questions. In *Proceedings of the Sixth Conference on Machine Translation*, pages 495–506, Online. Association for Computational Linguistics.

Mateusz Krubiński, Erfan Ghadery, Marie-Francine Moens, and Pavel Pecina. 2021b. MTEQA at WMT21 metrics shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1024–1029, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pretraining for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Arle Lommel, Aljoscha Burchardt, and Hans Uszkoreit. 2014. Multidimensional quality metrics (mqm): A framework for declaring and describing translation quality metrics. *Tradumàtica: tecnologies de la traducció*, 0:455–463.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020. Results of the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Ricardo Rei, Ana C Farinha, Chrysoula Zerva, Daan van Stigt, Craig Stewart, Pedro Ramos, Taisiya Glushkova, André F. T. Martins, and Alon Lavie. 2021. Are references really needed? unbabel-IST 2021 submission for the metrics shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1030–1040, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Thibault Sellam, Amy Pu, Hyung Won Chung, Sebastian Gehrmann, Qijun Tan, Markus Freitag, Dipanjan Das, and Ankur Parikh. 2020. Learning to evaluate translation beyond English: BLEURT submissions to the WMT metrics 2020 shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 921–927, Online. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Michal Stefanik, Vít Novotný, and Petr Sojka. 2021. Regressive ensemble for machine translation quality evaluation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1041–1048, Online. Association for Computational Linguistics.

Kosuke Takahashi, Yoichi Ishibashi, Katsuhito Sudoh, and Satoshi Nakamura. 2021. Multilingual machine translation evaluation metrics fine-tuned on pseudo-negative examples for WMT 2021 metrics task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1049–1052, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

A WMT 2021 System Mapping

- ‡: cushLEPOR(LM) (Han et al., 2021);
- $\perp$: C-SPEC and C-SPECpn (Takahashi et al., 2021);
- $\wedge$: tgt-regEMT and tgt-regEMT-baseline (Stefanik et al., 2021);
- $\|$: COMET-MQM_2021 and COMET-QE-MQM_2021-src (Rei et al., 2021);
- $\vee$: TER (Snover et al., 2006);
- †: BLEU (Papineni et al., 2002);
- $\top$: MTEQA (Krubiński et al., 2021a,b).

COMET-22:
Unbabel-IST 2022 Submission for the Metrics Shared Task

Ricardo Rei[*1,2,4], **José G. C. de Souza**[1], **Duarte M. Alves**[1,4],
Chrysoula Zerva[3,4] **Ana C Farinha**[1], **Taisiya Glushkova**[3,4],
Alon Lavie[1], **Luisa Coheur**[2,4], **André F. T. Martins**[1,3,4]
[1]Unbabel, Lisbon, Portugal, [2]INESC-ID, Lisbon, Portugal
[3]Instituto de Telecomunicações, Lisbon, Portugal
[4]Instituto Superior Técnico, University of Lisbon, Portugal

Abstract

In this paper, we present the joint contribution of Unbabel and IST to the WMT 2022 Metrics Shared Task. Our primary submission – dubbed COMET-22 – is an ensemble between a COMET estimator model trained with Direct Assessments and a newly proposed multitask model trained to predict sentence-level scores along with OK/BAD word-level tags derived from Multidimensional Quality Metrics error annotations. These models are ensembled together using a hyper-parameter search that weights different features extracted from both evaluation models and combines them into a single score. For the reference-free evaluation we present COMETKIWI. Similarly to our primary submission, COMETKIWI is an ensemble between two models. A traditional predictor-estimator model inspired by OPENKIWI and our new multitask model trained on Multidimensional Quality Metrics which can also be used without references. Both our submissions show improved correlations compared to state-of-the-art metrics from last year as well as increased robustness to critical errors.

1 Introduction

Automatic metrics for Machine Translation (MT) are a fundamental component of MT research and development. While human evaluation is still of great importance, automatic metrics allow the rapid evaluation and comparison of MT systems on large collections of text and facilitate expansion to low resource languages and domains. Neural fine-tuned metrics in particular, have shown the ability to leverage large multilingual data during training to better compare and assess the quality of state-of-the-art MT models, outperforming traditional lexical-based metrics. Hence, our research is targeted to and guided by the advancements in these metrics.

This paper presents the joint contribution of Unbabel and Instituto Superior Técnico (IST) to the WMT 2022 Metrics Shared Task (Freitag et al., 2022). We participated in the segment-level and system-level tracks, as well as the "QE-as-a-Metric". Similar to our participation last year (Rei et al., 2021), our models are based on the COMET framework[1] (Rei et al., 2020a).

Our efforts this year built on findings and observations from our participation in the WMT 2021 Metrics Shared Task (Rei et al., 2021; Freitag et al., 2021b) to further improve COMET for the Metrics task and to increase its robustness to translation errors such as deviation from named entities, reverse polarity and negation, deviation in numbers, etc. These types of fine-grained critical errors have been shown to be challenging for state-of-the-art metrics and QE systems (Amrhein and Sennrich, 2022; Kanojia et al., 2021). For that reason we aim to take advantage of finer-grained information, incorporating word-level supervision from Multidimensional Quality Metrics (MQM) annotations when available. This approach is motivated by the observed improvements in performance in WMT 2021 Metrics when fine-tuning on MQM data. Additionally, the importance of word-level supervision as an auxiliary task was established via our participation in the WMT 2022 Quality Estimation task (Zerva et al., 2022), where we found that we get a boost of performance across language pairs when we combine word- and sentence-level targets (Rei et al., 2022).

Overall, our main contributions are:

- We propose a new model architecture that is trained with a multitask objective to predict a sentence-level score along with word-level tags. This architecture is well suited for MQM data which comes in the form of sentence-level scores

[*] ✉ ricardo.rei@unbabel.com

[1] Code and models available at: https://github.com/Unbabel/COMET

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 578 - 585
December 7-8, 2022 ©2022 Association for Computational Linguistics

alongside the annotation spans. Also, similarly to UNITE (Wan et al., 2022) we can use this architecture with and without access to a reference translation.

- We show that ensembling scores from different models, trained with different annotations (e.g DA and MQM) can lead to improved correlations and more robust metrics.
- We corroborate our findings from last year (Rei et al., 2021) showing that reference-free evaluation is becoming competitive with reference-based evaluation.

Our submitted metrics, compared to two of the best submissions from last year, improve by a considerable margin in terms of correlations with MQM (≈ 4% in Kendall-Tau at segment level and ≈ 6% on system-level accuracy) and in the ability of detecting critical errors (≈ 30% in accuracy on `SMAUG` challenge set (Alves et al., 2022), a newly proposed challenge set built to test the robustness of MT metrics to errors in named entities, numbers, meaning, inserted content and missing content.).

2 Corpora

Every year, since 2017, the organisers of WMT News translation tasks collect annotations in the form of Direct Assessments (DA) (Graham et al., 2013). Recently, Freitag et al. (2021a) showed that DA annotations, collected by non-professional crowd-source workers, are noisy and unfit to measure the quality of high performing MT systems. For high quality MT evaluation the authors suggest the use of MQM annotations performed by professionals; they released annotations for English→German and Chinese→English on the WMT 2020 translation outputs. Since then, along with the DA annotations coming from the News translation task, the metrics task organizers collect additional MQM data for English→German (`en-de`), Chinese→English (`zh-en`), and English→Russian (`en-ru`) to evaluate metrics against a more reliable ground-truth.

With that said, to test the performance of our new systems we will use the MQM annotations from 2021 News domain. For training we will use all DA ranging from 2017 to 2020 and the remaining MQM annotations.

One of the findings from last years shared task is that metrics struggle to accurately penalize translations with errors in reversing negation (Freitag et al., 2021b). Also, Amrhein and Sennrich (2022) showed that using COMET as a utility function for Minimum Bayes Risk decoding is more likely to lead to errors in named entities and numbers when compared to lexical metrics such as CHRF. In order to measure progress on capturing those errors we will test our new metrics on `SMAUG` (Alves et al., 2022).

3 Implemented Systems

Our goal this year is to build more robust **C**ross-lingual **O**ptimized **M**etrics for **E**valuation of **T**ranslations by ensembling systems that model different aspects of MT evaluation. For that purpose we used three different systems: a COMET Estimator (Rei et al., 2020a) trained on DA, a newly proposed Sequence Tagger, trained with MQM data, that works with and without references, and a COMETKIWI (Rei et al., 2022) model trained on DA.

3.1 COMET Estimator

For a more comprehensive description of the Estimator architecture we direct the reader to the original paper (Rei et al., 2020a). Compared to our COMET-DA model from last year (Rei et al., 2021) we only changed hyper-parameters in order to maximize Kendall-Tau correlations with the MQM annotations from 2021 News domain.

3.2 QE Predictor-Estimator Model

For a more comprehensive description of the implemented Predictor-Estimator architecture we direct the reader to our QE system description paper (Rei et al., 2022). In summary, for this year QE shared task we combine the strengths of COMET and OPENKIWI, leading to models that adopt COMET training features, useful for multilingual generalization, along with the predictor-estimator architecture of OPENKIWI.

3.3 Extending COMET for Sequence Tagging:

Following our experiments for the Quality Estimation shared task we implemented a multitask COMET model that is trained to perform sequence tagging along with sentence-level regression.

Inspired by UNITE (Wan et al., 2022), our model receives three inputs:

1. Source-only (src): machine translated sentence concatenated with its source sentence.

		zh-en		en-de		en-ru			
	Nº Segments	9750		8959		8432			
	Correlations	ρ	τ	ρ	τ	ρ	τ	Avg. ρ	Avg. τ
Baselines	BLEU	0.215	0.153	0.086	0.065	0.123	0.094	0.141	0.104
	CHRF	0.116	0.088	0.116	0.088	0.213	0.165	0.192	0.143
	BLEURT	0.456	0.331	0.309	0.236	0.345	0.267	0.370	0.278
	COMET-20	0.463	0.336	0.270	0.206	0.330	0.256	0.355	0.266
	COMET-21	0.513	0.377	0.309	0.237	0.345	0.263	0.389	0.292
Primary Sub.	COMET-22	0.537	0.395	**0.366**	**0.281**	**0.407**	**0.315**	**0.437**	**0.330**
	MQM Sequence Tagger								
	$\hookrightarrow \hat{y}_{\text{tags}}$	0.311	0.222	0.302	0.237	0.362	0.314	0.325	0.258
	$\hookrightarrow \hat{y}_{\text{src}}$	0.487	0.356	0.347	0.266	0.359	0.276	0.398	0.299
	$\hookrightarrow \hat{y}_{\text{ref}}$	0.535	0.394	0.358	0.275	0.386	0.297	0.427	0.322
	$\hookrightarrow \hat{y}_{\text{uni}}$	**0.538**	**0.396**	0.360	0.277	0.382	0.294	0.427	0.322
	DA Estimator	0.495	0.362	0.289	0.221	0.369	0.285	0.384	0.289
QE metric	COMETKIWI	0.471	0.343	0.348	0.266	0.366	0.283	0.395	0.297
	MQM Sequence Tagger								
	$\hookrightarrow \hat{y}_{\text{tags}}$	0.431	0.312	0.279	0.218	0.332	0.257	0.313	0.245
	$\hookrightarrow \hat{y}_{\text{src}}$	0.283	0.201	0.347	0.266	0.310	0.268	0.348	0.262
	DA Pred-Estimator	0.487	0.356	0.286	0.219	0.359	0.276	0.377	0.284

Table 1: Segment-level Spearman R (ρ) and Kendall-Tau (τ) correlations for zh-en, en-de and en-ru 2021 MQM annotations for the News Domain.

2. Reference-only (ref): machine translated sentence concatenated with its reference.

3. Unified input (uni): machine translated sentence concatenated with both source and reference.

These inputs can be seen as a sequence with two parts: 1) the machine translated sentence $\boldsymbol{t} = \langle t_1, ..., t_n \rangle$ and 2) additional support information such as source and/or reference $\hat{\boldsymbol{r}} = \langle r_1, ..., r_m \rangle$.

Given that, for each input, our model works exactly like COMETKIWI. We run three forward passes and we store the corresponding sentence-level scores $\hat{y}_{\text{src}}$, $\hat{y}_{\text{ref}}$ and $\hat{y}_{\text{uni}}$. Additionally, we average the obtained word-level logits to derive a single sequence S of $\{\text{OK}, \text{BAD}\}$ tags from which we compute an additional sentence score by using a similar formula to MQM:

$$\hat{y}_{\text{tags}} = 1 - \frac{w \times \sum_{i}^{N_S} \mathbb{1}[S_i = \text{BAD}]}{N_S} \qquad (1)$$

where w is a severity penalty for BAD tags which we set to 1.

In sum, after running our new model we obtain 4 different scores: $\hat{y}_{\text{tags}}$, $\hat{y}_{\text{src}}$, $\hat{y}_{\text{ref}}$, $\hat{y}_{\text{uni}}$ which we can combine into a single quality score. Also, since this model is trained with a reference-less input it can be used, during inference, as a QE system. In those cases we run a single forward pass with the reference-less input and instead of 4 quality score we only get 2 ($\hat{y}_{\text{tags}}$ and $\hat{y}_{\text{src}}$).

Training Data. Since the MQM training data is not abundant and only covers 3 language pairs we start by training the above model without word-level information for 2 full epochs on DA ranging the shared task data from 2017 to 2020. Then, we fine-tune the model using the multitask setting described above with the available MQM training data for zh-en, en-de and en-ru.

3.4 Primary Submission

Our primary submission is an ensemble between a COMET Estimator model trained on top of XLM-R using DA from 2017 to 2020 and a sequence tagging model, such as the one described above, trained on top of InfoXLM (Chi et al., 2021). The final score is computed by a weighted average of the model outputs (5 scores), where the weights for each language pair were tuned with Optuna (Akiba et al., 2019)[2].

3.5 QE-as-a-metric Submission

Our primary submission is an ensemble between a COMETKIWI model trained on top of RemBERT (Chung et al., 2021) and the same sequence tagger from the primary submission but using a reference-less input during inference. The final score is computed by a weighted average in the

[2] We tuned weights specifically for the 3 MQM language pairs. For all other language pairs the weights were tuned by concatenating the MQM annotations for all three language pairs

same way as for our primary submission but using only the obtained 3 reference-less scores.

		zh-en	en-de	en-ru	avg.
	Nº Systems	15	17	16	
Baselines	BLEU	45.71	66.91	46.66	53.10
	CHRF	43.81	65.44	55.00	54.75
	BLEURT	48.57	83.09	70.83	67.50
	COMET-20	53.33	74.26	64.17	63.92
	COMET-21	53.33	78.68	70.83	67.61
Primary Sub.	COMET-22	64.76	**86.76**	70.83	74.12
	MQM Sequence Tagger				
	$\hookrightarrow \hat{y}_{\text{tags}}$	60.95	80.88	**75.83**	72.55
	$\hookrightarrow \hat{y}_{\text{src}}$	**72.38**	**86.76**	70.83	76.66
	$\hookrightarrow \hat{y}_{\text{ref}}$	71.42	**86.76**	**75.83**	**78.00**
	$\hookrightarrow \hat{y}_{\text{uni}}$	69.52	85.29	72.5	75.77
	DA Estimator	50.47	71.32	66.66	62.82
QE metric	COMETKIWI	68.57	86.02	70.83	75.14
	MQM Sequence Tagger				
	$\hookrightarrow \hat{y}_{\text{tags}}$	47.61	**86.76**	59.16	64.51
	$\hookrightarrow \hat{y}_{\text{src}}$	67.61	78.68	72.50	72.93
	DA Pred-Estimator	**72.38**	57.35	70.83	66.85

Table 2: System-level accuracy for `zh-en`, `en-de` and `en-ru` 2021 MQM annotations for the News Domain.

4 Experimental Results

As we have seen in Section 2, for our experiments we use WMT 2021 News MQM annotations from last year shared task (Freitag et al., 2021b) for testing our metrics. As for baselines we used lexical metrics such as CHRF (Popović, 2015) and BLEU (Papineni et al., 2002) and three state-of-the-art metrics: BLEURT (Sellam et al., 2020; Pu et al., 2021), COMET-20 (Rei et al., 2020b) and COMET-21 (Rei et al., 2021)[3].

4.1 Segment-level

Segment-level correlations for 2021 MQM annotations on the News domain are shown in Table 1. We used both Spearman R (ρ) and Kendall-Tau (τ) correlation metrics to evaluate our models.

From this table we can observe that some individual scores from the MQM Sequence Tagger already outperform state-of-the-art metrics such as BLEURT and (COMET-20/21). Also, our newly trained DA Estimator is able to outperform BLEURT achieving results close to COMET-21 without ever seeing MQM data. Finally, when ensembled together, we are able to improve correlations by $\approx 1\%$ for both reference-based and reference-free metrics.

4.2 System-level

System-level results for 2021 MQM annotations for the News domain are shown in Table 2. To evaluate how our metrics perform we used the pairwise accuracy proposed in (Kocmi et al., 2021), which simulates a real world scenario where we are interested in comparing two systems and deciding which one is better.

Similarly to segment-level results, from Table 2, we can observe that the accuracy of individual scores from the MQM Sequence Tagger outperform, on average, strong baselines such as BLEURT and COMET-20/21. Nonetheless, when ensembled together, these scores do not improve the overall accuracy which seems to be obtained by using the MQM Sequence Tagger with references only. Another interesting finding is that our QE submission (COMET-KIWI) achieves higher accuracy than our primary submission COMET-22. Also, depending on the language pair the best accuracy is either achieved by $\hat{y}_{\text{src}}$ or $\hat{y}_{\text{ref}}$ but not by $\hat{y}_{\text{uni}}$. This seems to indicate that the unified score is not learning to take the best out of the source and reference and that there might be a best way to combine these two signals.

4.3 Robustness to Critical Errors

The `SMAUG` challenge set was built to specifically test the robustness of metrics in capturing 5 different phenomena: deviation in Named Entities (NE), deviation in Numbers (NUM), deviation in meaning (MEAN), insertion of content (INS) and removal of content (DEL). The goal of this challenge set is to check if metrics correctly penalize an incorrect translation that was created by perturbing a reference. To do so, the perturbed translation (t) is scored using source (s) and reference (r) against an alternative reference ($\hat{r}$). The goal of a metric f is to score $\hat{r}$ above t ($f(s, \hat{r}, r) > f(s, t, r)$). To measure f's performance we will look at the accuracy over the entire challenge set for each phenomena:

$$Acc^P = \frac{\sum_i^{N_P} \mathbb{1}[f(s, \hat{r}, r) > f(s, t, r)]}{N_P} \qquad (2)$$

where P denotes a phenomena and N_P the number of examples for that specific phenomena.

[3] For all neural fine-tuned metrics we used the checkpoints that were used as primary submissions for the WMT20 and WMT21 Metric tasks, more precisely, `BLEURT20`, `wmt20-comet-da` and `wmt21-comet-mqm`

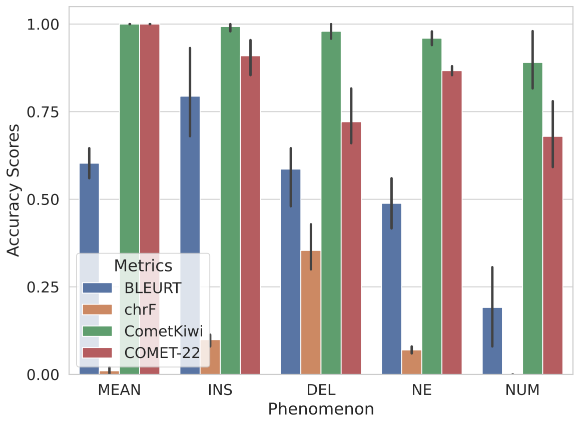


Figure 1: Accuracy Scores on the SMAUG Challenge Set for the baseline and submitted metrics.

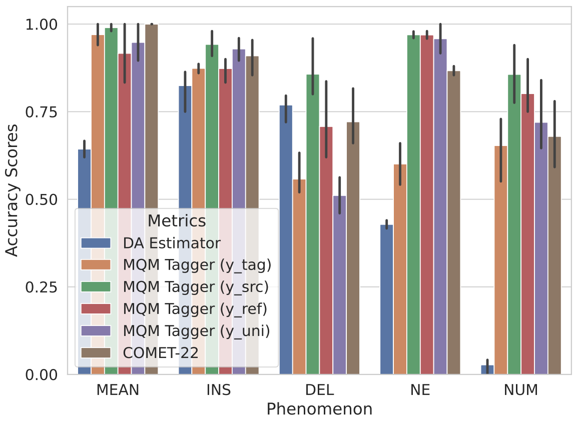


Figure 2: Accuracy Scores on the SMAUG Challenge Set for Primary Submission and respective individual scores.

Figure 1 presents the accuracy of our submissions against a lexical baseline (CHRF) and a learnt baseline (BLEURT) [4]. From these figure we can observe that our reference-free submission seems to be more robust than our primary submission which indicates that, when the reference is present, models look at lexical overlap and can be oblivious to critical errors that were derived from small perturbations. Also, we can observe that our submissions achieve a perfect accuracy on detecting deviations in meaning (which tests phenomena such as negation), above 0.65 accuracy in detecting wrong numbers and above 0.86 accuracy in detecting incorrect named entities. All of which were not correctly detected by previous state-of-the-art metrics such as BLEURT and COMET-20/21.

We also compare the performance of each ensemble with the individual models that compose it. In Figure 2, we observe that the DA Estimator has the worst overall performance. Also, the MQM Sequence Tagger $\hat{y}_{\text{src}}$ achieves the highest scores over all individual models, further suggesting that reference-free evaluation is more robust to these errors. Our final submission, while not reaching the highest accuracy for all phenomena, obtains good results in all cases. Regarding our QE-as-a-metric submission, Figure 3 shows that both the ensemble and individual systems achieve very high scores. Our submission outperforms the MQM Taggers and obtains a performance similar to the DA Predictor-Estimator.

5 Related work

For years, **classic n-gram matching** MT evaluation metrics such as BLEU (Papineni et al., 2002) have been adopted by the MT community as a primary form of MT evaluation yet, recently, these classic metrics have been outperformed by metrics based on large pretrained models such as BERT (Bojar et al., 2017; Ma et al., 2018, 2019; Mathur et al., 2020).

Metrics based on large pretrained models can be divided into two categories: 1) **Embedding-distance metrics** and, 2) **Fine-tuned metrics**. **Embedding-distance metrics** replaced the typical word/n-gram matching by fuzzy matches based on dense representations. Examples of such metrics are BERTSCORE (Zhang et al., 2020) and YISI-1 (Lo, 2019), which has been a top performing metric since WMT 2019 Metrics task (Ma et al., 2019). Note that these metrics used the embedding models without any further fine-tuning relying only on their ability to capture semantic similarity. On the other hand, **fine-tuned metrics** such as RUSE (Shimanaka et al., 2018), BLEURT (Sellam et al., 2020) and COMET (Rei et al., 2020a) modify the underlying embedding models in order to learn how to produce quality scores such as DA and/or MQM, and thus to achieve higher correlations with human judgements of MT quality.

Recently, the evaluation of metrics has been extended to consider not only correlations with human judgements but also sensitivity to specific errors in translations. Namely, several works focused on translations with critical errors, which Specia et al. (2021) defines as translations that deviate in

[4] Performance of COMET-20 and 21 is similar to the performance shown by BLEURT while BLEU accuracy is close to 0.

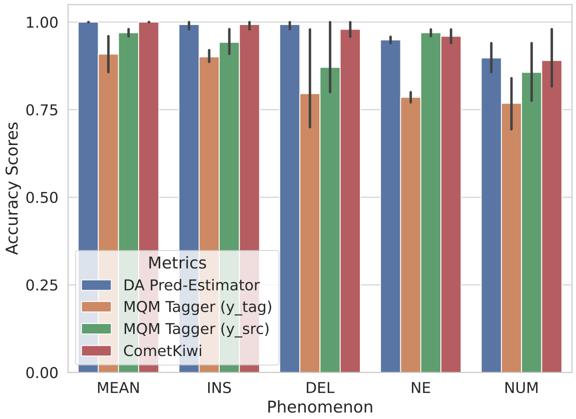


Figure 3: Accuracy Scores on the SMAUG Challenge Set for QE-as-a-metric Submission and respective individual scores.

meaning from their source in such way that they are misleading and can carry health, safety, legal, reputation, religious or financial implications. Amrhein and Sennrich (2022) show that COMET is less sensitive to errors in named entities and numbers than CHRF; Freitag et al. (2021b) found that several metrics struggle with negation and sentiment polarity errors; and Kanojia et al. (2021) showed that several reference-free metrics fail to detect errors related to omitting negation markers.

6 Conclusions

We present the joint contribution of Unbabel and IST to the WMT 2022 Metrics shared task. We propose a new architecture trained in a multitask setting which takes advantage of sentence-level scores along with supervision from MQM annotation spans. Inspired by UNITE, our new model can be used with and without references showing promising results when references are not available.

Our primary submission ensembles our new model along with the COMET Estimator architecture showing both higher correlations and improved robustness to phenomena that was deemed challenging in previous shared task editions.

Finally, our "QE-as-a-metric" submission yet again has shown that reference-free is competitive to reference-based evaluation not only at segment-level but also at system-level and in terms of detecting critical errors.

Acknowledgements

This work was supported by the P2020 programs (MAIA, contract 045909), by EU's Horizon Europe Research and Innovation Actions (UTTER, contract 101070631), by the European Research Council (ERC StG DeepSPIN 758969), and by the Fundação para a Ciência e Tecnologia through contract UIDB/50008/2020.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Duarte M. Alves, Ricardo Rei, Ana C. Farinha, José G. C. de Souza, and André F. T. Martins. 2022. Robust MT evaluation with Sentence-level Multilingual data Augmentation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Chantal Amrhein and Rico Sennrich. 2022. Identifying Weaknesses in Machine Translation Metrics Through Minimum Bayes Risk Decoding: A Case Study for COMET. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, Online. Association for Computational Linguistics.

Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017. Results of the WMT17 metrics shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 489–513, Copenhagen, Denmark. Association for Computational Linguistics.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. Association for Computational Linguistics.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking Embedding Coupling in Pre-trained Language Models. In *International Conference on Learning Representations*.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021a. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. Results of WMT22 Metrics Shared Task: Stop Using BLEU – Neural Metrics Are Better and More Robust. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021b. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Diptesh Kanojia, Marina Fomicheva, Tharindu Ranasinghe, Frédéric Blain, Constantin Orăsan, and Lucia Specia. 2021. Pushing the right buttons: Adversarial evaluation of quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 625–638, Online. Association for Computational Linguistics.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Qingsong Ma, Ondřej Bojar, and Yvette Graham. 2018. Results of the WMT18 metrics shared task: Both characters and embeddings achieve good performance. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 671–688, Belgium, Brussels. Association for Computational Linguistics.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020. Results of the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Amy Pu, Hyung Won Chung, Ankur Parikh, Sebastian Gehrmann, and Thibault Sellam. 2021. Learning compact metrics for MT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 751–762, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ricardo Rei, Ana C Farinha, Chrysoula Zerva, Daan van Stigt, Craig Stewart, Pedro Ramos, Taisiya Glushkova, André F. T. Martins, and Alon Lavie. 2021. Are references really needed? unbabel-IST 2021 submission for the metrics shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1030–1040, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020a. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020b. Unbabel's participation in the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 911–920, Online. Association for Computational Linguistics.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C. Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte M. Alves, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. CometKiwi: IST-Unbabel 2022 Submission for the Quality Estimation Shared Task. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2018. RUSE: Regressor using sentence embeddings for automatic machine translation evaluation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 751–758, Belgium, Brussels. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the WMT 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 684–725, Online. Association for Computational Linguistics.

Yu Wan, Dayiheng Liu, Baosong Yang, Haibo Zhang, Boxing Chen, Derek Wong, and Lidia Chao. 2022. UniTE: Unified translation evaluation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8117–8127, Dublin, Ireland. Association for Computational Linguistics.

Chrysoula Zerva, Frédéric Blain, Ricardo Rei, Piyawat Lertvittayakumjorn, José G. C. de Souza, Steffen Eger, Diptesh Kanojia, Duarte Alves, Constantin Orăsan, Marina Fomicheva, André F. T. Martins, and Lucia Specia. 2022. Findings of the WMT 2022 Shared Task on Quality Estimation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Alibaba-Translate China's Submission for WMT 2022 Metrics Shared Task

Yu Wan[1,2*] **Keqin Bao**[1,3*] **Dayiheng Liu**[1] **Baosong Yang**[1] **Derek F. Wong**[2]
Lidia S. Chao[2] **Wenqiang Lei**[4] **Jun Xie**[1]
[1]DAMO Academy, Alibaba Group [2]NLP[2]CT Lab, University of Macau
[3]University of Science and Technology of China [4]National University of Singapore
nlp2ct.ywan@gmail.com baokeqin@mail.ustc.edu.cn
{liudayiheng.ldyh,yangbaosong.ybs,qingjing.xj}@alibaba-inc.com
{derekfw,lidiasc}@um.edu.mo wenqianglei@gmail.com

Abstract

In this report, we present our submission to the WMT 2022 Metrics Shared Task. We build our system based on the core idea of **UNITE** (**Uni**fied **T**ranslation **E**valuation), which unifies source-only, reference-only, and source-reference-combined evaluation scenarios into one single model. Specifically, during the model pre-training phase, we first apply the pseudo-labeled data examples to continuously pre-train UNITE. Notably, to reduce the gap between pre-training and fine-tuning, we use data cropping and a ranking-based score normalization strategy. During the fine-tuning phase, we use both Direct Assessment (DA) and Multidimensional Quality Metrics (MQM) data from past years' WMT competitions. Specially, we collect the results from models with different pre-trained language model backbones, and use different ensembling strategies for involved translation directions.

1 Introduction

Translation metric aims at delivering accurate and convincing predictions to identify the translation quality of outputs with access to one or many gold-standard reference translations (Ma et al., 2018, 2019; Mathur et al., 2020; Freitag et al., 2021b). As the development of neural machine translation research (Vaswani et al., 2017; Wei et al., 2022), the metric methods should be capable of evaluating the high-quality translations at the level of semantics rather than surfance-level features (Sellam et al., 2020; Ranasinghe et al., 2020; Rei et al., 2020; Wan et al., 2022a). In this paper, we describe Alibaba Translate China's submissions to the WMT 2022 Metrics Shared Task to deliver a more adequate evaluation solution at the level of semantics.

Pre-trained language models (PLMs) like BERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) have shown promising results in identifying the quality of translation outputs. Compared to conventional statistical- (*e.g.*, BLEU, Papineni et al., 2002 and representation-based methods (*e.g.*, BERTSCORE, Zhang et al., 2020), the model-based approaches (*e.g.*, BLEURT, Sellam et al., 2020; COMET, Rei et al., 2020; UNITE, Wan et al., 2022a) show their strong ability on delivering more accurate quality predictions, especially those approaches which apply source sentences as additional input for the metric model (Rei et al., 2020; Takahashi et al., 2020; Wan et al., 2021, 2022a). Specifically, those metric models are designed as a combination of PLM and feedforward network, where the former is in charge of deriving representations on input sequence, and the latter predicts the translation quality based on the representation. The metric model, which is trained on synthetic or human annotations following a regressive objective, learns to mimic human predictions to identify the translation quality of the hypothesis sentence.

Although those model-based metrics have shown promising results in modern applications and translation quality estimation, they still show their own shortcomings as follows. First, they often handle one specific evaluation scenario, *e.g.*, COMET serves source-reference-only evaluation, where the source and reference sentence should be concurrently fed to the model for prediction. For the other evaluation scenarios, they hardly give accurate predictions, showing the straits of metric models due to the disagreement between training and inference. Besides, recent studies have investigated the feasibility of unifying those evaluation scenarios into one single model, which can further improve the evaluation correlation with human ratings in any scenario among source-only, reference-only, and source-reference-combined evaluation (Wan et al., 2021, 2022a). This indicates that, training with multiple input formats than a specific one can deliver more appropriate predictions for translation

[*] Equal contribution. Work was done when Yu Wan and Keqin Bao were interning at DAMO Academy, Alibaba Group.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 586 - 592
December 7-8, 2022 ©2022 Association for Computational Linguistics

quality identification. More importantly, unifying all translation evaluation functionalities into one single model can serve as a more convenient toolkit in real-world applications.

Following the idea of Wan et al. (2022a) and the experience in previous competition (Wan et al., 2021), we directly use the pipeline of UniTE (Wan et al., 2022a) to build models for this year's metric task. Each of our models can integrate the functionalities of source-only, reference-only, and source-reference-combined translation evaluation into itself. When collecting the system outputs for the WMT 2022 Metrics Shared Task, we employ our UniTE models to predict the translation quality scores following the source-reference-combined setting. Compared to the previous version of UniTE (Wan et al., 2022a), we reform the synthetic training set for the continuous pre-training phase, raising the ratio of training examples consisting of high-quality hypothesis sentences. Also, during fine-tuning our metric model, we apply available Direct Assessment (DA, Bojar et al., 2017; Ma et al., 2018, 2019; Mathur et al., 2020) and Multidimensional Quality Metrics datasets (MQM, Freitag et al., 2021a,b) from previous WMT competitions to further improve the performance of our model. Specifically, for each translation direction among English to German (En-De), English to Russian (En-Ru), and Chinese to English (Zh-En) directions, we applied different ensembling strategies to achieve a better correlation with human ratings on MQM 2021 dataset. Results on WMT 2021 MQM dataset further demonstrate the effectiveness of our method.

2 Method

As outlined in §1, we apply the UniTE framework (Wan et al., 2022a) to obtain metric models. We use three types of input formats (*i.e.*, source-only, reference-only, and source-reference-combined) during training. While during inference, we only use the source-reference-combined paradigm to collect evaluation scores. In this section, we introduce the applied model architecture (§2.1), synthetic data construction method (§2.2), and model training strategy (§2.3) for this year's metric competition.

2.1 Model architecture

Input Format Following Wan et al. (2022a), we construct the input sequence for source-only, reference-only, and source-reference-combined input formats as follows:

$$\mathbf{x}_{\text{SRC}} = \texttt{[BOS]}\mathbf{h}\texttt{[DEL]}\mathbf{s}\texttt{[EOS]}, \quad (1)$$

$$\mathbf{x}_{\text{REF}} = \texttt{[BOS]}\mathbf{h}\texttt{[DEL]}\mathbf{r}\texttt{[EOS]}, \quad (2)$$

$$\mathbf{x}_{\text{SRC+REF}} = \texttt{[BOS]}\mathbf{h}\texttt{[DEL]}\mathbf{s}\texttt{[DEL]}\mathbf{r}\texttt{[EOS]}, \quad (3)$$

where [BOS], [DEL] and [EOS] represent the beginning, the delimeter, and the ending of sequence,[1] and $\mathbf{h}$, $\mathbf{s}$, and $\mathbf{r}$ are hypothesis, source, and reference sentence, respectively. During the pre-training phase, we applied all input formats to enhance the performance of UniTE models.

Model Backbone Selection Aside from the reference sentence which is written in the same language as the hypothesis sentence, the source is in another different language. We believe that, cross-lingual semantic alignments can ease the model training on source-only and source-reference-combined scenarios. Referring to the setting of existing methods (Ranasinghe et al., 2020; Rei et al., 2020; Sellam et al., 2020; Wan et al., 2022a), they apply XLM-R (Conneau et al., 2020) as the backbone of evaluation models for better multilingual support. In this competition, we additionally use INFOXLM (Chi et al., 2021), which enhances the XLM-R model with cross-lingual alignments, as the backbone of our UniTE models.

Model Training Following Wan et al. (2022a), we first equally split all examples into three parts, each of which only serves one input format training. As to each training example, after concatenating the required input sentences into one sequence and feeding it to PLM, we collect the corresponding representations – $\mathbf{H}_{\text{REF}}, \mathbf{H}_{\text{SRC}}, \mathbf{H}_{\text{SRC+REF}}$ for each input format, respectively. After that, we use the output embedding assigned with CLS token $\mathbf{h}$ as the sequence representation. Finally, a feedforward network takes $\mathbf{h}$ as input and gives a scalar p as a prediction. Taking $\mathbf{x}_{\text{SRC}}$ as an example:

$$\mathbf{H}_{\text{SRC}} = \texttt{PLM}(\mathbf{x}_{\text{SRC}}) \in \mathbb{R}^{(l_h+l_s)\times d}, \quad (4)$$

$$\mathbf{h}_{\text{SRC}} = \texttt{CLS}(\mathbf{H}_{\text{SRC}}) \in \mathbb{R}^{d}, \quad (5)$$

$$p_{\text{SRC}} = \texttt{FeedForward}(\mathbf{h}_{\text{SRC}}) \in \mathbb{R}^{1}, \quad (6)$$

where l_h and l_s are the lengths of $\mathbf{h}$ and $\mathbf{s}$, respectively.

[1] Those symbols may vary if we use different PLMs, *e.g.*, "[BOS]", "[SEP]", and "[SEP]" for English BERT (Devlin et al., 2019), and "<s>", "</s> </s>", and "</s>" for XLM-R (Conneau et al., 2020).

For learning objectives, we apply the mean squared error (MSE) as the loss function:

$$\mathcal{L}_{\text{SRC}} = (p_{\text{SRC}} - q)^2, \tag{7}$$

where q is the given ground-truth score. Note that, the batch size is the same across all input formats to avoid the training imbalance. During each update, the final learning objective is the sum of losses for all formats:

$$\mathcal{L} = \mathcal{L}_{\text{REF}} + \mathcal{L}_{\text{SRC}} + \mathcal{L}_{\text{SRC+REF}}. \tag{8}$$

2.2 Synthetic Data Construction

To better enhance the translation evaluation ability, we first construct a synthetic dataset for continuous pre-training. The overall stage for obtaining the dataset consists of the following steps: 1) collecting synthetic data from parallel data provided by the WMT Translation task; 2) downgrading the translation quality and keeping the consistency of synthetic and MQM datasets; 3) relabeling them with a ranking-based scoring strategy.

Collecting Synthetic Data Specifically, we first conduct parallel data from this year's WMT Translation competition as the source-reference sentence pairs Then, we obtain hypothesis sentences via translating the source using online translation engines, *e.g.*, `Google Translate`[2] and `Alibaba Translate`[3].

Quality Downgrading We follow existing works (Sellam et al., 2020; Wan et al., 2022a) to apply the word/span dropping strategy to downgrade the quality of hypothesis sentences, thus increasing the ratio of training examples consisting of bad translation outputs. Specially, we notice that the translation quality of hypothesis sentences in the MQM dataset is rather higher than that in the DA dataset. In practice, to reduce the translation quality distribution gap between the synthetic and MQM datasets, we randomly select 15% examples of the entire dataset, which is lower than the applied ratio (*i.e.*, 30%) in BLEURT (Sellam et al., 2020) and UNITE (Wan et al., 2022a).

Data Labeling After downgrading the translation quality of synthetic hypothesis sentences, we then collect predicted scores for each triple as the learning supervision. To increase the confidence of pseudo-labeled scores, we use multiple UNITE checkpoints trained with different random seeds to label the synthetic data. Besides, to reduce the gap of predicted scores among different translation directions, we applied the ranking-based scoring strategy as in Wan et al. (2022a).

2.3 Training Pipeline

Pre-train with Synthetic Data First, we use the synthetic dataset to continuously pre-train our UNITE models to enhance the evaluation ability on three input formats.

Fine-tune with DA Dataset After training UNITE models on the synthetic dataset, we apply the DA dataset for the first stage of model fine-tuning. Considering the support of multilingual translation evaluation, we collect all DA datasets from the previous years, and we leave the year 2021 out of training due to the reported bug from the official committee. We think that, although the DA and MQM datasets have different scoring rules, training UNITE models on DA as an additional phase can enhance both the model robustness and the support of multilingualism. Besides, the number of examples in the DA dataset is extremely larger than that in MQM. The training examples from the DA dataset can provide more learning signals for UNITE model training.

Fine-tune with MQM Dataset After fine-tuning UNITE models on the DA dataset, we then apply the MQM dataset for the second stage of model fine-tuning. For this year's competition, we first use MQM 2020 dataset during this stage, and testify the performance of our models on MQM 2021 to tune the hyper-parameters. Then, after identifying the hyper-parameters, we use all MQM datasets to fine-tune, choose two models whose backbones are XLM-R and INFOXLM, and collect the ensembled scores as submissions.

2.4 Model Ensembling

For each training pipeline, we use the three random seeds to train UNITE models. However, when identifying the performance of all models on the MQM 2021 dataset, we find it hard to select the same strategy across all domains and translation directions. In practice, we select the models trained with different random seeds for each translation direction.

[2] https://translate.google.com

[3] https://translate.alibaba.com

3 Experiments

3.1 Experiment Settings

Implementations All of our models are implemented with the released UNITE repository.[4] We choose the large version of XLM-R (Conneau et al., 2020) and INFOXLM (Chi et al., 2021) as the PLM backbones of all UNITE models, and directly use the released checkpoints from Huggingface Transformers (Wolf et al., 2020).[5]

Continuous pre-training Following Wan et al. (2022a), we collect the translation hypotheses from 10 directions, *i.e.*, English-Czech/German/Japanese/Russian/Chinese, as those translation directions are engaged with massive parallel datasets and the performance of corresponding online translation engines is relatively high. For each translation direction, we collect 0.5M hypotheses, and label the translation quality scores as described in §2.2.

Hyper-parameters Following the setting in Wan et al. (2022a), the feedforward network of our UNITE model contains three linear transition layers, whose output dimensionalities are 3,072, 1,024, and 1, respectively. Between any two adjacent layers, the hyperbolic tangent is arranged as the activations. During the continuous pre-training phase, we set the batch size for each input format as 1,024, and tune the hyper-parameters for our models. For the models whose backbone is XLM-R, the learning rates for PLM and feedforward network are $1.0 \cdot 10^{-4}$ for PLM, and $3.0 \cdot 10^{-4}$, respectively. For the models whose backbone is INFOXLM, the learning rates are $5.0 \cdot 10^{-5}$ for PLM, and $1.5 \cdot 10^{-4}$, respectively. For all the fine-tuning steps, we use the batch size as 32 across all settings, and the learning rates for PLM and feedforward network are $5.0 \cdot 10^{-6}$ for PLM, and $1.5 \cdot 10^{-5}$, respectively.

Performance Evaluation Following the previous setting (Ma et al., 2018, 2019; Mathur et al., 2020; Freitag et al., 2021b), we use the variant Kendall's Tau to evaluate the performance of our models on the MQM 2021 dataset. For comparison, we directly use the officially released COMET checkpoints (Rei et al., 2020)[6], and select the checkpoints which are trained with DA or MQM datasets.

Results Conduction When collecting the results for submitting predictions, we ensembled the models by directly averaging the predictions on the same example. We do not apply the idea of uncertainty-aware sampling (Zhou et al., 2020; Wan et al., 2020; Glushkova et al., 2021) during inference, because it takes far more additional time to collect the results.

4 Results and Analysis

Baselines The experimental results are conducted in Table 1. As seen, among all involved baselines, the source-only evaluation models (models marked with "QE") perform worse than their corresponding source-reference-combined ones, dropping 7.2 and 7.4 Kendall's Tau correlation on DA and MQM settings. This verifies that, the reference sentence in model translation quality evaluation offers more information for metric models to help deliver accurate predictions (Rei et al., 2020; Takahashi et al., 2020; Wan et al., 2022a). Besides, the model fine-tuned on the DA dataset performs slightly better than that on MQM. We think that the DA dataset may show its advantage in the robustness of multilingual support and the scale of the training dataset.

UNITE models As to our UNITE models, replacing the XLM-R backbone with INFOXLM PLM for metric models does not deliver consistent improvement on average. Specifically, for both News and TED domains, the UNITE model with INFOXLM as the backbone shows a better correlation on En-De direction, whereas worse on En-Ru and Zh-En than XLM-R. In addition, the COMET-DA-2021 performs best in En-Ru direction, where we think the reason lies in the scarcity of En-Ru training examples in MQM. In practice, during collecting the ensembled outputs, we mainly use the $\text{UNITE}_{\text{INFOXLM}}$ models for En-De, and $\text{UNITE}_{\text{XLM-R}}$ for En-Ru and Zh-En.

5 Conclusion

In this paper, we describe our submission UNITE for the sentence-level Metrics Shared Task at WMT 2022. We apply UNITE (Wan et al., 2022a) as the pipeline of our models. During training, we utilize three input formats to train our models on our synthetic, DA, and MQM data sequentially. Besides,

[4] https://github.com/wanyu2018umac/UniTE

[5] https://huggingface.co/xlm-roberta-large, https://huggingface.co/microsoft/infoxlm-large

[6] https://github.com/Unbabel/COMET/

Model	News			TED			All
	En-De	En-Ru	Zh-En	En-De	En-Ru	Zh-En	
COMET-QE-DA-2021	23.7	34.6	8.3	12.3	22.5	8.5	14.4
COMET-DA-2021	28.1	**43.1**	15.2	20.2	28.5	15.9	21.6
COMET-QE-MQM-2021	26.7	33.3	6.7	10.6	22.3	5.5	12.8
COMET-MQM-2021	27.5	42.5	11.4	18.5	28.8	13.3	20.2
UNITE$_{\text{XLM-R}}$	27.7	39.0	**16.3**	19.7	**31.2**	**17.3**	**25.3**
UNITE$_{\text{INFOXLM}}$	**40.0**	36.2	13.0	**25.3**	28.7	9.2	24.9

Table 1: Kendall's Tau correlation (%) on MQM 2021 dataset. The best results for each translation direction are bold. Taking XLM-R as backbone shows better result on En-Ru and Zh-En, and INFOXLM on En-De.

we ensemble the two models which consist of two different backbones – XLM-R and INFOXLM. Experiments demonstrate the reliability of our model for identifying the quality of translation outputs, whereas the two models whose backbones XLM-R and INFOXLM show different performance for different translation directions.

For the future work, we think that exploring the feasibility of model-based evaluation metrics for other natural language processing tasks is interesting. We believe that, building reliable evaluation metrics for translation diversity (Lin et al., 2022, 2021), domain-specific translation quality (Yao et al., 2020; Wan et al., 2022b), and natural language generation (Liu et al., 2022; Yang et al., 2021, 2022) is also of vital importance for the natural language processing community.

Besides, we also submit the source-only predictions of our models to this year's WMT Quality Estimation Shared Task, achieving 1st place on multilingual and En-Ru, and 2nd place on En-De and Zh-En sub-tracks. This further demonstrates the effectiveness of our UNITE approach, that unifying all evaluation scenarios into one single model can enhance the model performances on all evaluation tasks. We believe that, the idea of unifying three kinds of translation evaluation functionalities (*i.e.*, source-only, reference-only, and source-reference-combined) into one single model can deliver strong evaluation models on all scenarios. This research topic is worth further exploration in the future.

Acknowledgements

The participants would like to send great thanks to the committee and the organizers of the WMT Metrics Shared Task competition. Besides, the authors would like to thank the reviewers and meta-review for their insightful suggestions.

This work was supported in part by the Science and Technology Development Fund, Macau SAR (Grant No. 0101/2019/A2), the Multi-year Research Grant from the University of Macau (Grant No. MYRG2020-00054-FST), National Key Research and Development Program of China (No. 2018YFB1403202), and Alibaba Group through Alibaba Innovative Research (AIR) Program.

References

Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017. Results of the WMT17 metrics shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 489–513, Copenhagen, Denmark. Association for Computational Linguistics.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages

4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021a. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021b. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Taisiya Glushkova, Chrysoula Zerva, Ricardo Rei, and André F. T. Martins. 2021. Uncertainty-aware machine translation evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3920–3938, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Huan Lin, Baosong Yang, Liang Yao, Dayiheng Liu, Haibo Zhang, Jun Xie, Min Zhang, and Jinsong Su. 2022. Bridging the gap between training and inference: Multi-candidate optimization for diverse neural machine translation. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2622–2632, Seattle, United States. Association for Computational Linguistics.

Huan Lin, Liang Yao, Baosong Yang, Dayiheng Liu, Haibo Zhang, Weihua Luo, Degen Huang, and Jinsong Su. 2021. Towards user-driven neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4008–4018, Online. Association for Computational Linguistics.

Xin Liu, Dayiheng Liu, Baosong Yang, Haibo Zhang, Junwei Ding, Wenqing Yao, Weihua Luo, Haiying Zhang, and Jinsong Su. 2022. Kgr4: Retrieval, retrospect, refine and rethink for commonsense generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11029–11037.

Qingsong Ma, Ondřej Bojar, and Yvette Graham. 2018. Results of the WMT18 metrics shared task: Both characters and embeddings achieve good performance. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 671–688, Belgium, Brussels. Association for Computational Linguistics.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020. Results of the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. TransQuest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5070–5081, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Kosuke Takahashi, Katsuhito Sudoh, and Satoshi Nakamura. 2020. Automatic machine translation evaluation using source language inputs and cross-lingual language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3553–3558, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yu Wan, Dayiheng Liu, Baosong Yang, Tianchi Bi, Haibo Zhang, Boxing Chen, Weihua Luo, Derek F. Wong, and Lidia S. Chao. 2021. RoBLEURT submission for WMT2021 metrics task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1053–1058, Online. Association for Computational Linguistics.

Yu Wan, Dayiheng Liu, Baosong Yang, Haibo Zhang, Boxing Chen, Derek Wong, and Lidia Chao. 2022a. UniTE: Unified translation evaluation. In *Proceedings of the 60th Annual Meeting of the Association* *for Computational Linguistics (Volume 1: Long Papers)*, pages 8117–8127, Dublin, Ireland. Association for Computational Linguistics.

Yu Wan, Baosong Yang, Derek F. Wong, Yikai Zhou, Lidia S. Chao, Haibo Zhang, and Boxing Chen. 2020. Self-paced learning for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1074–1080, Online. Association for Computational Linguistics.

Yu Wan, Baosong Yang, Derek Fai Wong, Lidia Sam Chao, Liang Yao, Haibo Zhang, and Boxing Chen. 2022b. Challenges of neural machine translation for short texts. *Computational Linguistics*, 48(2):321–342.

Xiangpeng Wei, Heng Yu, Yue Hu, Rongxiang Weng, Weihua Luo, and Rong Jin. 2022. Learning to generalize to more: Continuous semantic augmentation for neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7930–7944, Dublin, Ireland. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Kexin Yang, Wenqiang Lei, Dayiheng Liu, Weizhen Qi, and Jiancheng Lv. 2021. POS-Constrained Parallel Decoding for Non-autoregressive Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5990–6000, Online. Association for Computational Linguistics.

Kexin Yang, Dayiheng Liu, Wenqiang Lei, Baosong Yang, Haibo Zhang, Xue Zhao, Wenqing Yao, and Boxing Chen. 2022. GCPG: A general framework for controllable paraphrase generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4035–4047, Dublin, Ireland. Association for Computational Linguistics.

Liang Yao, Baosong Yang, Haibo Zhang, Boxing Chen, and Weihua Luo. 2020. Domain transfer based data augmentation for neural query translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4521–4533, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yikai Zhou, Baosong Yang, Derek F. Wong, Yu Wan, and Lidia S. Chao. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6934–6944, Online. Association for Computational Linguistics.

Quality Estimation via Backtranslation at the WMT 2022 Quality Estimation Task

Sweta Agrawal*
University of Maryland
sweagraw@umd.edu

Nikita Mehandru*
University of California, Berkeley
nmehandru@berkeley.edu

Niloufar Salehi
University of California, Berkeley
nsalehi@berkeley.edu

Marine Carpuat
University of Maryland
marine@umd.edu

Abstract

This paper describes submission to the WMT 2022 Quality Estimation shared task (Task 1: sentence-level quality prediction, Zerva et al. (2022)). We follow a simple and intuitive approach: estimating MT quality by automatically back-translating hypotheses into the source language using a multilingual MT system. Using standard MT evaluation metrics, we then compare the resulting backtranslation with the original source. We find that even the best-performing backtranslation-based scores perform substantially worse than supervised QE systems, including the organizers' baseline. However, combining backtranslation-based metrics with off-the-shelf QE scorers improves correlation with human judgments, suggesting that they can indeed complement a supervised QE system.

1 Introduction

Sophisticated approaches to MT quality estimation (QE) based on large pre-trained models and careful training regimen have enabled great progress in recent years. However, when using online MT systems, such QE technology is not yet available to users and backtranslation provides an appealingly simple strategy to estimate translation quality whether by humans or by automated sytems. Lay users often rely on backtranslation to assess MT quality in languages that they do not understand (Somers, 2005; Mehandru et al., 2022). As a result, from a user experience standpoint, using backtranslation for QE is easy to explain. Furthermore, with the increasing popularity of multilingual neural MT systems that can easily translate between multiple language pairs in any direction, backtranslations are very cheap to obtain, since they do not even require training an auxiliary MT system in the reverse translation direction.

However, the effectiveness of backtranslation for estimating the quality of MT remains unclear. In early rule-based and statistical MT systems, Somers (2005) shows that, when using automatic evaluation methods (e.g., BLEU), backtranslation cannot discriminate good MT systems from bad ones, nor between texts that are easy or hard to translate. This led him to conclude that "round trip translation [is] good for nothing". Recently, Moon et al. (2020) revisited the use of backtranslation for QE with neural systems for MT and with embedding-based similarity metrics to enable a more sophisticated comparison of the backtranslation with the source. They obtained strong results on the WMT 2019 QE task, outperforming the YISI-2 metric (Lo, 2019) on system-level evaluations, but exhibited rather low correlations on the segment-level task which is more directly aligned with how humans use BT to gauge MT quality.

The goal of our submission is to pitch a backtranslation-based QE score that can complement state-of-the-art quality estimation systems in the controlled settings of the WMT shared task (Zerva et al., 2022) and understand its reliability as a sentence-level quality estimation technique.

2 Approach

Following Moon et al. (2020), given a source sentence x and a MT hypothesis, we translate y back into the source language using an off-the-shelf multilingual model M, yielding backtranslation $\tilde{x}$. We then compare x and $\tilde{x}$ using standard machine translation evaluation metrics, and hypothesize that the distance between x and $\tilde{x}$, referred to as **BT-score(x, $\tilde{x}$),** can be an indicative of the translation quality of y.

However, MT systems are prone to making errors and are shown to hallucinate content. When the BT system makes an error, it can misguide the users in believing that the translation is a) erroneous when it is not and b) correct when the BT system magically recovers the source content. In order to improve the reliability of the BT-based QE

[*] equal contribution.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 593 - 596
December 7-8, 2022 ©2022 Association for Computational Linguistics

BT Metrics	**Footprint** Bytes	**Params.**	**Development Set** Pearson	 Spearman	**Test Set** Pearson	 Spearman
BLEU	0	0	0.179	0.170	0.141	0.137
chrF	0	0	0.203	0.181	0.184	0.174
BERTScore	0	177853440	0.292	0.296	0.325	0.285
Baseline[1]	2280011066	564527011	n/a	n/a	0.560	0.576

Table 1: Pearson and Spearman correlation between backtranslation-based QE metrics and Direct Assessment judgments on the WMT 2022 En-Cs task.

Metrics	**En-Cs (DA)** Dev	 Test	**En-Ru (MQM)** Dev	 Test	**Zh-En (MQM)** Dev	 Test
[1] BT-BERTScore	0.296	0.285	0.262	0.210	0.151	0.249
[2] Comet-Src	0.461	0.519	0.505	0.383	0.213	0.223
Multiply([1], [2])	**0.467**	**0.523**	**0.512**	**0.390**	**0.216**	**0.257**
Baseline[2]	n/a	0.560	n/a	0.330	n/a	0.164

Table 2: Spearman correlation between QE metrics and human judgments on the WMT 2022 Sentence Level Quality Estimation task: Combining BT-BERTScore and Comet-Src improves correlation with human judgments across the board.

metrics, **BT-score**(x, $\tilde{x}$), and to understand whether they can complement off-the-shelf QE scorers that directly estimate the quality of a source sentence and a MT hypothesis, **FT-score**(x, y), we also propose to combine the two evaluation methods using a simple multiplication ("AND") operation.

Back-translation Model The backward translations were generated from Facebook's mBART-50 Many-to-One and One-to-Many multilingual machine translation (MMT) models. The MMT model can translate between 49 languages into and out of English, and uses 12 layers with 1,024 sized embeddings, 4,096 feedforward neural network (FNN) embedding dimensions, and 16 heads for both encoder and decoders.[3]

MT Evaluation Metrics We experiment with model-free and model-based evaluation metrics. We apply the following sentence-level scores to compare detokenized backtranslations $\tilde{x}$ with the source x:

- BLEU: we use the Sacrebleu implementation of sentence-level BLEU, with an exponential decay smoothing.[4] (Papineni et al., 2002)
- chrF: we use the Sacrebleu implementation of the chrF score, which takes a maximum character n-gram order count of six and calculates the number of ngram overlap between hypothesis and reference n-grams. (Popović, 2015)
- BERTScore: we compute the F-score based on wordpiece-level embedding similarities of, weighted by inverse document frequency (idf), using BERT as the embedding model (Zhang et al., 2019).[5].

We use the publicly available QE metric, Comet-Src ("wmt21-comet-qe-mqm") to compute FT-score(x, y).

3 Official Results using BT-based Metrics

We evaluate our approach on the English-Czech sentence-level quality prediction subtask. As our approach is unsupervised, we do not use the training data provided by the organizers. We report results obtained on the development and test sets, using the Pearson and Spearman correlations with human judgments of quality.

[3] https://huggingface.co/facebook/mbart-large-50-many-to-one-mmt/, https://huggingface.co/facebook/mbart-large-50-one-to-many-mmt/

[4] https://github.com/mjpost/sacrebleu

[5] https://pypi.org/project/bert-score/

	$DA >= -1$	$DA < -1$	$DA >= 0$	$DA < 0$	$DA >= 1$	$DA < 1$
BT-BERTScore	0.197	**0.230**	0.133	0.222	0.022	0.235
Comet-Src	**0.397**	0.139	**0.337**	**0.313**	**0.139**	**0.413**

Table 3: En-Cs segment-level correlation in different quality buckets according to the direct assessment scores.

Sample	**Development Set**			
	z-mean	BT-BLEU	BT-chrF	BT-BERT
Source: Arif Lohar briefly went into acting in punjabi movies before returning to his music career at the age of 22 . **Output:** Arif Lohar krátce začal hrát v Punjabi filmech , než se v roce 22 vrátil ke své hudební kariéře . **BT Source:** Arif Lohar briefly began acting in Punjabi films before returning to his musical career in the year 22.	-1.486	20.95	62.57	0.949
Source: Promulgate Thai Royal and noble titles back and return the title to politician who was canceled . **Output:** Promulgate Thajské královské a šlechtické tituly zpět a vrátit titul politici , který byl zrušen . **BT Source:** Promulgate Thai royal and noble titles back and return the title of politician that was abolished.	-1.781	48.34	73.94	0.959
Source: Ika-6 na utos ; re - runs ; aired on gma life tv for the first time ; replacing I heart davao . **Output:** Ika-6 na utos ; re - runs ; poprvé vysíláno na gma life TV ; nahrazuje I heart davao . **BT Source:** Ika-6 on utos; re-runs; first broadcast on gma life TV; replaces I heart davao.	-2.935	18.00	53.63	0.941

Table 4: Three randomly sampled sentences from the bottom 5% according to DA scores.

As can be seen in Table 4, BERTScore provides a better correlation with human judgments than BLEU and chrF consistently on the development and test sets. This is expected since the underlying BERT model provides a more semantically informed comparison than n-gram metrics. However, the backtranslation metrics yield low correlation scores overall, underperforming the organizer's baseline on the test set.

Our results are complementary to Moon et al. (2020) in that they suggest that BT-based metrics might be better suited to ranking diverse outputs from systems of varying overall quality, than those from a single MT system, i.e. at predicting quality assessments at the segment level.

4 Can BT-based scorers complement existing QE metrics?

While standalone evaluation using BT-based scoring significantly lags behind supervised SOTA QE baselines, we evaluate whether BT-based metrics can provide reliable complementary judgments to a supervised off-the-shelf QE scorer in Table 2. We combine the best BT-based scorer, BT-BERTscore and a standard QE scorer, Comet-Src using a simple multiplication operation. On three sentence level quality estimation tasks: En-Cs (DA), En-Ru (MQM) and Zh-En (MQM), combining both BT and QE scores result in improved correlation across the board over individual metrics, outperforming baselines on both En-Ru and Zh-En.

In order to better understand the source of this improvement, we divide the En-Cs development dataset into different buckets based on the direct assessment scores and report correlation on the resulsubsets in Table 3. On very bad quality translations, i.e. $DA <= -1$, BT-BERTScore exhibits a higher correlation than Comet-Src, suggesting that it is able to more reliably distinguish between bad translations than Comet-Src, hence complementing the QE metric.

5 Qualitative Analysis on En-Cs

In Table 2, we randomly sampled three sentences from the lowest 5% of the human direct assessment scores from the development set data and report the corresponding BT-BLEU, BT-chrF, and BT-BERTScores. The outputs depict how the forward translation output can be of poor quality, as indicated by the human direct assessment scores. However, the semantic similarity between the source and the back-translated source can still suggest that the forward translation is correct. When we apply machine translation to other domains, this can be problematic and misleading since users may mistakenly impart higher trust levels when using backtranslation techniques. From the same table, we can also observe that the automatic metrics cannot capture salient errors as suggested by the high scores generated by the automatic metric for the second example ("who was canceled" vs "that was abolished"). This finding is in line with prior work that has shown a positive correlation between *human evaluations* conducted on input sentences and translated outputs with *human evaluations* on input sentences and round-trip sentences (Aiken and Park, 2010). These results together call for a more systematic assessment of the role of backtranslation in lay users perceptions of MT quality.

6 Conclusion

We evaluated backtranslation-based unsupervised quality estimation systems on the sentence-level quality estimation task. Our results show that backtranslation bases scorers fall substantially behind supervised models such as the organizers' baseline. However, they can complement off-the-shelf QE metrics in distinguishing bad translations. Qualitative analysis on En-Cs indicates that while backtranslation can be a poor indicator of translation quality, the automatic metrics derived using the source and the backtranslated source might also add to the unreliability of the scorer. This suggests that more investigation is needed to determine whether backtranslation can be used effectively for QE in practical systems, whether for automatic quality estimation or to provide quality feedback to human users.

References

Milam Aiken and Mina Park. 2010. The efficacy of round-trip translation for mt evaluation. *Translation Journal*, 14(1):1–10.

Chi-kiu Lo. 2019. YiSi - a Unified Semantic MT Quality Evaluation and Estimation Metric for Languages with Different Levels of Available Resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Nikita Mehandru, Samantha Robertson, and Niloufar Salehi. 2022. Reliable and safe use of machine translation in medical settings. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (Seoul, South Korea)(FAccT'22). Association for Computing Machinery, New York, NY, USA.*

Jihyung Moon, Hyunchang Cho, and Eunjeong L. Park. 2020. Revisiting round-trip translation for quality estimation. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 91–104, Lisboa, Portugal. European Association for Machine Translation.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.

Harold Somers. 2005. Round-trip Translation: What Is It Good For? In *Proceedings of the Australasian Language Technology Workshop 2005*, pages 127–133, Sydney, Australia.

Chrysoula Zerva, Frederic Blain, Ricardo Rei, Piyawat Lertvittayakumjorn, José G. C. de Souza, Steffen Eger, Diptesh Kanojia, Duarte Alves, Constantin Orăsan, Marina Fomicheva, Andre F. T. Martins, and Lucia Specia. 2022. Findings of the wmt 2022 shared task on quality estimation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675.*

Alibaba-Translate China's Submission for WMT 2022 Quality Estimation Shared Task

Keqin Bao[1,2*] **Yu Wan**[1,3*] **Dayiheng Liu**[1] **Baosong Yang**[1] **Wenqiang Lei**[4]
Xiangnan He[2] **Derek F. Wong**[3] **Jun Xie**[1]
[1]DAMO Academy, Alibaba Group [2]University of Science and Technology of China
[3]NLP^{2}CT Lab, University of Macau [4]National University of Singapore
baokeqin@mail.ustc.edu.cn nlp2ct.ywan@gmail.com
{liudayiheng.ldyh,yangbaosong.ybs,qingjing.xj}@alibaba-inc.com
wenqianglei@gmail.com xiangnanhe@gmail.com derekfw@um.edu.mo

Abstract

In this paper, we present our submission to sentence-level MQM benchmark at Quality Estimation Shared Task, named **UNITE** (**Uni**fied **T**ranslation **E**valuation). Specifically, our systems employ the framework of UNITE, which combined three types of input format during training with a pre-trained language model. First, we apply the pseudo-labeled data examples for the continuously pre-training phase. Notably, to reduce the gap between pre-training and fine-tuning, we use data pruning and a ranking-based score normalization strategy. For the fine-tuning phase, we use both Direct Assessment (DA) and Multidimensional Quality Metrics (MQM) data from past years' WMT competitions. Finally, we collect the source-only evaluation results, and ensemble the predictions generated by two UNITE models, whose backbones are XLM-R and INFOXLM, respectively. Results show that our models reach 1st overall ranking in the Multilingual and English-Russian settings, and 2nd overall ranking in English-German and Chinese-English settings, showing relatively strong performances in this year's quality estimation competition.

1 Introduction

Quality Estimation (QE) aims at evaluating machine translation without access to a gold-standard reference translation (Blatz et al., 2004; Specia et al., 2018). Different from other evaluation tasks (*e.g.*, metric), QE arranges its process of evaluation via only accessing source input. As the performance of modern machine translation approaches increase (Vaswani et al., 2017; Lin et al., 2022; Wei et al., 2022; Zhang et al., 2022), the QE systems should better quantify the agreement of cross-lingual semantics on source sentence and translation hypothesis. The evaluation paradigm of QE shows its own potential for real-world applications (Wang et al., 2021; Park et al., 2021; Specia et al., 2021). This paper describes Alibaba Translate China's submission to the sentence-level MQM benchmark at WMT 2022 Quality Estimation Shared Task (Zerva et al., 2022).

In recent years, pre-trained language models (PLMs) have shown their strong ability on extracting cross-lingual information (Conneau et al., 2020; Chi et al., 2021). To achieve a higher correlation with human ratings on the quality of translation outputs, plenty of trainable model-based QE approaches appear, *e.g.*, COMET-QE (Rei et al., 2020) and QEMIND (Wang et al., 2021). They both first derive the embeddings assigned with source and hypothesis sentence with given PLM, then predict the overall score based on their embeddings with a followed feedforward network. Those model-based approaches have greatly facilitated the development of the QE community. However, those models can only handle source-only input format, which neglects the other two evaluation scenarios, *i.e.*, reference-only and source-reference-combined evaluation. More importantly, training with multiple input formats can achieve a higher correlation with human assessments than individually training on specific evaluation scenarios (Wan et al., 2021, 2022a). Those findings indicate that, the QE and Metric tasks share plenty of knowledge when identifying the quality of translated outputs, and unifying the functionalities of three evaluation scenarios into one model can also enhance the performance of the evaluation model on each scenario.

As a consequence, when building a single model for a sentence-level QE task, we use the pipeline of UNITE (Wan et al., 2022a), which integrates source-only, reference-only, and source-reference-combined translation evaluation ability into one single model. When collecting the system outputs for WMT 2022 Quality Estimation Shared Task, we employ our UNITE models to predict

[*] Equal contribution. Work was done when Keqin Bao and Yu Wan were interning at DAMO Academy, Alibaba Group.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 597 - 605
December 7-8, 2022 ©2022 Association for Computational Linguistics

the translation quality scores following a source-only setting. As for the training data, we collect synthetic data examples as supervision for continuous pre-training and apply a dataset pruning strategy to increase the translation quality of the training set. Also, during fine-tuning our QE model, we use all available Direct Assessment (DA, Bojar et al., 2017; Ma et al., 2018, 2019; Mathur et al., 2020) and Multidimensional Quality Metrics datasets (MQM, Freitag et al., 2021a,b) from previous WMT competitions to further improve the performance of our model. Besides, regarding the applied PLM for UniTE models, we find that for English-Russian (En-Ru) and Chinese-English (Zh-En) directions, PLM enhanced with cross-lingual alignments (InfoXLM, Chi et al., 2021) can deliver better results than conventional ones (XLM-R, Conneau et al., 2020). Moreover, for each subtask including English to German (En-De), En-Ru, Zh-En, and multilingual direction evaluations, we build an ensembled QE system to derive more accurate and convincing results as final predictions.

Our models show impressive performances in all translation directions. When only considering the primary metric – Spearman's correlation, we get 2nd, 3rd, and 3rd place in En-Ru, Zh-En, and multilingual direction, respectively. More notably, when taking all metrics into account, despite the slight decrease in Spearman's correlations, our systems show outstanding overall performance than other systems, achieving 1st place in En-Ru and multilingual, and 2nd in En-De and Zh-En direction.

2 Method

As outlined in §1, we apply the UniTE framework (Wan et al., 2022a) to obtain QE models. We unify three types of input formats (*i.e.*, source-only, reference-only, and source-reference-combined) into one single model during training. While during inference, we only use the source-only paradigm to collect evaluation scores. In this section, we introduce the applied model architecture (§2.1), synthetic data construction method (§2.2), and model training strategy (§2.3).

2.1 Model architecture

Input Format Following Wan et al. (2022a), we design our QE model which is capable of processing **source-only**, **reference-only**, and **source-reference-combined** evaluation scenarios. Consequently, for the consistency of training across all input formats, we construct the input sequence for source-only, reference-only, and source-reference-combined input formats as follows:

$$\mathbf{x}_{\text{SRC}} = \langle \mathrm{s} \rangle \mathbf{h} \langle / \mathrm{s} \rangle \langle / \mathrm{s} \rangle \mathbf{s} \langle / \mathrm{s} \rangle, \tag{1}$$

$$\mathbf{x}_{\text{REF}} = \langle \mathrm{s} \rangle \mathbf{h} \langle / \mathrm{s} \rangle \langle / \mathrm{s} \rangle \mathbf{r} \langle / \mathrm{s} \rangle, \tag{2}$$

$$\mathbf{x}_{\text{SRC+REF}} = \langle \mathrm{s} \rangle \mathbf{h} \langle / \mathrm{s} \rangle \langle / \mathrm{s} \rangle \mathbf{s} \langle / \mathrm{s} \rangle \langle / \mathrm{s} \rangle \mathbf{r} \langle / \mathrm{s} \rangle, \tag{3}$$

where $\mathbf{h}$, $\mathbf{s}$, and $\mathbf{r}$ represent hypothesis, source, and reference sentence, respectively. During the pre-training phase, we apply all input formats to enhance the performance of QE models. Notably, we only use the source-only format setting when fine-tuning on this year's dev set and inferring the test set.

Model Backbone Selection The core of quality estimation aims at evaluating the translation quality of output given source sentence. As the source and hypothesis sentence are from different languages, evaluating the translation quality requires the ability of multilingual processing. Furthermore, we believe that those PLMs which possess cross-lingual semantic alignments can ease the learning of translation quality evaluation.

Referring to the setting of existing methods (Ranasinghe et al., 2020; Rei et al., 2020; Sellam et al., 2020; Wan et al., 2022a), they often apply XLM-R (Conneau et al., 2020) as the backbone of evaluation models for better multilingual support. To testify whether cross-lingual alignments can help the evaluation model training, we further apply InfoXLM (Chi et al., 2021), which enhances the XLM-R model with cross-lingual alignments, as the backbone of evaluation models.

Model Training For the training dataset including source, reference, and hypothesis sentences, we first equally split all examples into three parts, each of which only serves one input format training. As to each training example, after concatenating the required input sentences into one sequence and feeding it to PLM, we collect the corresponding representations – $\mathbf{H}_{\text{REF}}, \mathbf{H}_{\text{SRC}}, \mathbf{H}_{\text{SRC+REF}}$ for each input format, respectively. After that, we use the output embedding assigned with CLS token $\mathbf{h}$ as the sequence representation. Finally, a feedforward network takes $\mathbf{h}$ as input and gives a scalar p as a prediction. Taking $\mathbf{x}_{\text{SRC}}$ as an example:

$$\mathbf{H}_{\text{SRC}} = \texttt{PLM}(\mathbf{x}_{\text{SRC}}) \in \mathbb{R}^{(l_h+l_s)\times d}, \quad (4)$$

$$\mathbf{h}_{\text{SRC}} = \texttt{CLS}(\mathbf{H}_{\text{SRC}}) \in \mathbb{R}^{d}, \quad (5)$$

$$p_{\text{SRC}} = \texttt{FeedForward}(\mathbf{h}_{\text{SRC}}) \in \mathbb{R}^{1}, \quad (6)$$

where l_h and l_s are the lengths of $\mathbf{h}$ and $\mathbf{s}$, respectively.

For the learning objective, we apply the mean squared error (MSE) as the loss function:

$$\mathcal{L}_{\text{SRC}} = (p_{\text{SRC}} - q)^2, \quad (7)$$

where q is the given ground-truth score. Note that, when training on three input formats, one single step includes three substeps, each of which is arranged on one specific input format. Besides, the batch size is the same across all input formats to avoid the training imbalance. During each update, the final learning objective can be written as the sum of losses for each format:

$$\mathcal{L} = \mathcal{L}_{\text{REF}} + \mathcal{L}_{\text{SRC}} + \mathcal{L}_{\text{SRC+REF}}. \quad (8)$$

2.2 Constructing Synthetic Data

To better enhance the translation evaluation ability of pre-trained models, we first construct synthetic dataset for continuous pre-training (Wan et al., 2022a). The pipeline for obtaining such dataset consists of the following steps: 1) collecting synthetic data from parallel data provided by the WMT Translation task; 2) labeling samples with a ranking-based scoring strategy; 3) pruning data samples to increase the quality of dataset; 4) relabeling them with a ranking-based scoring strategy.

Collecting Synthetic Data Pseudo datasets for model pre-training has been proven effective for obtaining well-performed evaluation models (Sellam et al., 2020; Wan et al., 2021, 2022a). Moreover, as in Wan et al. (2022a), training on three input formats requires massive pseudo examples. Specifically, we first obtain parallel data from this year's WMT Translation task as the source-reference sentence pairs, and translate the source using online translation engines, *e.g.*, `Google Translate`[1] and `Alibaba Translate`[2], to generate the hypothesis sentence. As discussed in Sellam et al. (2020), the conventional pseudo hypotheses are usually of high translation quality. Consequently, the dataset hardly possesses a higher level of translation quality diversity, making it difficult to train evaluation models. We follow existing works (Wan et al., 2022a; Sellam et al., 2020) to apply the word and span dropping strategy to attenuate hypotheses quality, increasing the ratio of training examples consisting of bad translation outputs.

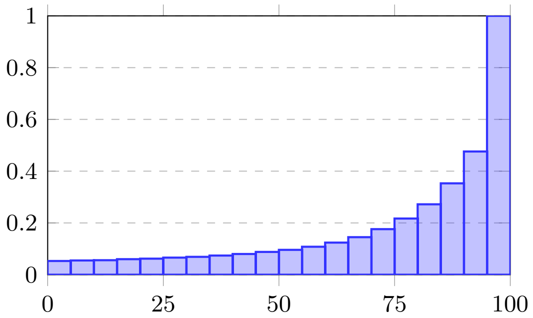


Figure 1: The cumulative distribution of scores in WMT 2020 and 2021 MQM datasets. The x-axis represents the annotated score while the y-axis represents the ratio.

Data Labeling and Pruning After downgrading the translation quality of synthetic hypothesis sentences, we then collect predicted scores for each triple as the learning supervision using checkpoint from UNITE (Wan et al., 2022a).[3] As discussed in Wan et al. (2022a) and Sellam et al. (2020), scores labeled by low-quality metrics have poor consistency, confusing the model learning during the training period. To increase the confidence of pseudo-labeled scores, we use multiple UNITE checkpoints trained with different random seeds to label the synthetic data (Wan et al., 2022a). Besides, to reduce the gap of predicted scores among different translation directions, as well as alleviate the bias among multiple evaluation approaches, we follow the scoring methods in UNITE (Wan et al., 2022a), using the idea of Borda count (Ho et al., 1994; Emerson, 2013). After sorting the collected prediction scores, we use their ranking indexes instead, and apply the conventional Z-score strategy to normalize them.

During our preliminary experiments, we find that the quality of hypotheses in the MQM 2020 and 2021 dataset is generally high. As shown in Figure 1, more than 64% of the human-annotated scores are higher than 90. To further mitigate the disagreement of translation quality distributions between pre-training and test datasets, we arrange

[1] `https://translate.google.com`

[2] `https://translate.alibaba.com`

[3] https://github.com/wanyu2018umac/UniTE

data pruning for synthetic data. Specifically, for each language pair, we ascendingly sort the synthetic examples by their scores, and split the examples into 5 bins. For the examples in each bin, we randomly drop 90%, 80%, 60%, 20%, and 0% data examples, yielding. We obtain 0.5M synthetic data for each language pair, and renormalize our prediction scores by the ranking-based manners as described before. In total, we collect pseudo examples on 10 translation directions, *i.e.*, English ↔ Czech/German/Japanese/Russian/Chinese, each of which contains 0.5M data tuples formatted as $\langle \mathbf{h}, \mathbf{s}, \mathbf{r}, q \rangle$.

2.3 Training Pipeline

To train UNITE models, the available datasets consist of synthetic examples (as in §2.2), human annotations (*i.e.*, DA and MQM), as well as provided development set for this year. In practice, we arrange the training pipeline into three steps as follows.

Pre-train with Synthetic Data As illustrated in §2.2, after collecting synthetic dataset, we use them to continuously pre-train our UNITE models to enhance the evaluation ability on three input formats.

Fine-tune with DA Dataset After collecting pre-trained checkpoints, we first fine-tune them with human-annotated DA datasets. Although the DA and MQM datasets have different scoring rules, training UNITE models on DA as an additional phase can enhance both the model robustness and the support of multilinguality. In practice, we collect all DA datasets from the year 2017 to 2020, yielding 853k training examples. Notably, we leave the year 2021 out of training due to the reported bug from the organizational committee.

Fine-tune with MQM Dataset For the evaluation test set which is assessed with MQM scoring rules, we arrange the MQM dataset from the year 2020 and 2021 for fine-tuning models at the end of the training phase, consisting of 75k examples. Specifically, during this step, we first use the provided development set to tune hyper-parameters for continuous pre-training and fine-tuning, and directly use all data examples to fine-tune our UNITE models following the previous setting.

2.4 Results Conduction

To select appropriate checkpoints, we evaluate our models on this year's development set and select top-3 models for each translation direction. Furthermore, to fully utilize the development set, we conduct a 5-fold cross-validation on the development set to select the best hyper-parameters for each top-3 model training on them. Finally, we use the best hyper-parameters to fine-tune one single model on the entire development set.

As to the results conduction, we first applied multiple random seeds for each setting, and select the checkpoint with the best performance for model training. Besides, to further increase the accuracy of ensembled scores, we choose two checkpoints whose backbones are XLM-R and INFOXLM, respectively.

Notably, uncertainty estimation has been verified in Machine Translation and Translation Evaluation communities (Wan et al., 2020; Zhou et al., 2020; Glushkova et al., 2021). However, applying this method is time consunming and we do not try it in this year's QE task.

3 Experiments

Experiment Settings We choose the large version of XLM-R (Conneau et al., 2020) and INFOXLM (Chi et al., 2021) as the PLM backbones of all UNITE models. The feedforward network contains three linear transition layers, whose output dimensionalities are 3,072, 1,024, and 1, respectively. Between any two adjacent layers, a hyperbolic tangent is arranged as the activations.

During the pre-training phase, we use the WMT 2021 MQM dataset as the development set to tune the hyper-parameters for continuous pre-training and DA fine-tuning phases. For the XLM-R setting, we apply the learning rate as $1.0 \cdot 10^{-5}$ for PLM, and $3.0 \cdot 10^{-5}$ for the feedforward network. Especially, for INFOXLM setting, we halve the corresponding learning rates to maintain the training stability. Besides, we find that raising the batch size can make the training more stable. In practice, we set the batch size for each input format as 1,024. For the following fine-tuning steps, we use the batch size as 32 across all settings.

Evaluation Setup As requested by organizers, we primarily evaluate our systems in terms of Spearman's correlation metric between the predicted scores and the human annotations for each translation direction. Apart from that, we also take other metrics, *e.g.*, Pearson's correlation, into account. Note that, during the evaluation of the multilingual phase, we directly calculate the correlation

Model	Multilingual	En-De	En-Ru	Zh-En
COMET-QE-21 (Zerva et al., 2021)	39.8	49.4	46.5	23.5
UniTE-pretrain	14.0	36.0	15.2	23.8
UniTE-pretrain-prune	28.5	41.5	22.2	20.4
UniTE-pretrain-prune + DA	**44.5**	49.3	50.3	25.2
UniTE-pretrain-prune + MQM	29.2	39.8	49.0	23.9
UniTE-pretrain-prune + DA + MQM	40.2	**52.3**	58.5	25.7
UniTE-InfoXLM-pretrain-prune + DA + MQM	32.2	47.7	**59.0**	**27.1**

Table 1: Spearman's correlaion (%) on this year's development dataset. The best result for each translation direction are bolded. Applying both DA and MQM datasets for fine-tuning can achieve better results. Taking XLM-R as backbone shows better result on En-De, and InfoXLM on Zh-En and En-Ru.

Model	Multilingual	En-De	En-Ru	Zh-En
Single model	41.1	46.1	47.4	31.3
5-fold ensembling	42.7	53.1	48.4	**34.7**
XLM-R + InfoXLM ensembling	**45.6**	**55.0**	**50.5**	33.6

Table 2: Spearman's correlaion (%) on this year's test set. The best results for each translation direction are viewed in bold. Using 5-fold ensembling strategy delivers better correlation on Zh-En translation direction, and ensembling models trained on different PLM backbones conducts better results on multilingual, En-De, and En-Ru setting.

score for all predictions instead of conducting that for each language direction individually.

Baseline We introduce COMET-QE-21 (Zerva et al., 2021), one of the best-performed QE models as our strong baseline. COMET-QE-21 have shown their strong performance in WMT 2021 QE (Specia et al., 2021) and Metrics Shared Task (Freitag et al., 2021b) competitions. We directly apply the official released COMET-21-QE baseline[4], and use the well-trained checkpoints to infer on this year's development set for comparison.

Main Results We first testify the effectiveness of our systems on this year's development set. As shown in Table 1, our models outperform COMET-QE-21 in all translation directions. As to the results of final submissions, we list the results in Table 2.

4 Analysis

In this section, we discuss the effectiveness of all strategies, *i.e.*, data pruning (§4.1), training data arrangement (§4.2), backbone selection (§4.3), and model ensembling methods(§4.4).

4.1 Data pruning

We first investigate the impact of the data pruning strategy in Table 1. When using the pruneped data to train UniTE models, the performance gains significant improvements, with 14.5, 5.5, and 7.0 Spearman's correlation on Multilingual, En-De, and En-Ru translation direction, respectively. As discussed in §2.2, most training examples in MQM dataset have a higher translation quality. The data pruning method can reduce the ratio of training examples that contains poorly translated hypotheses. In contrast to the unpruneped synthetic dataset, the ratio of those examples consisting of well-translated outputs is raised. Consequently, we can reduce the translation quality distribution gap between synthetic and MQM datasets, and continuous pre-training and fine-tuning phases can share a great deal of learned knowledge. The experimental results validate our thinking, that the data pruning strategy offers a higher transferability of quality evaluation from synthetic to MQM data examples, making the model learning easier on the latter.

4.2 Training Data

To identify which dataset among DA and MQM is more important during fine-tuning, we conduct an experiment for comparing the corresponding effectiveness. As shown in Table 1, using DA or MQM dataset can both give performance improvement compared to only using synthetic data. Notably, the combination of DA and MQM datasets can further

[4] https://github.com/Unbabel/COMET/

boost the performance in En-Ru/En-De/Zh-En directions. However, when comparing UNITE-DA-MQM to UNITE-DA, an unexpected performance drop in the Multilingual setting is observed.

We think the reasons behind this phenomenon are two-fold. On one hand, DA data has 34 translation directions, while MQM data only has three specific directions (*i.e.*, En-De, En-Ru, and Zh-En). The annotation rules applied for those two datasets are inconsistent with each other. Training the model on MQM data can boost the performance in a specific direction. While a model trained on DA data is possessed with a more general evaluation ability for more translation directions, thus delivering more stable results on multilingual evaluation scenarios. On the other hand, for MQM data items, even though the scores may be similar across translation directions and competition years, the corresponding translation quality may vary vastly. For example, a score of 0.3 may be relatively a high score in MQM 2021 Zh-En subset, while it is rather low in this year's En-De direction. This phenomenon is quite critical when handling examples from multiple translation directions. As scores from the involved two translation directions are not compatible, training on those examples concurrently may downgrade the multilingual performance of our models.

4.3 Backbone Selection

As in Table 1, UniTE-pretrain-prune + DA + MQM is trained with XLM-R backbone, while UNITE-INFOXLM-pretrain-prune + DA + MQM is trained with INFOXLM using the same hyper-parameters and strategy. As seen, after updating the backbone of UNITE model from XLM-R to INFOXLM, the latter model outperforms the former in En-Ru and Zh-En directions, with the improvement of Spearman's correlation at 0.5 and 1.4, respectively. We can see that the quality estimation model can benefit from the cross-lingual alignment knowledge during model training. However, as to the En-De direction, the performance shows a significant drop at 4.6. We attribute this to the reason, that English and German are from the same language family, where the two languages can obtain a great deal of cross-lingual knowledge via similar tokens with the same meaning. For Multilingual direction, we claim that the impact of training data makes it unconfident which has been discussed in §4.2.

4.4 Ensemble Methods

As in Table 2, the ensembled models show great improvement on all translation directions. The difference between XLM-R and INFOXLM lies in the training objective and applied training dataset. For the quality estimation task whose core lies in the semantic alignment across languages, the knowledge engaged inside those two PLM models can be complementary to each other. Except for Zh-En direction, XLM + INFOXLM ensembling outperforms the 5-fold ensembling method in three tracks, with the performance increase being 2.9, 1.9, and 2.1 for Multilingual, En-De, and En-Ru settings, respectively. This demonstrates that, ensembling models constructed with different backbones can give better results compared to the k-fold ensembling strategy.

5 Conclusion

In this paper, we describe our UNITE submission for the sentence-level MQM task at WMT 2022. We apply data pruning and a ranking-based scoring strategy to collect massive synthetic data. During training, we utilize three input formats to train our models on our synthetic, DA, and MQM data sequentially. Besides, we ensemble the two models which consist of two different backbones – XLM-R and INFOXLM. Experiments show that, our unified training framework can deliver reliable evaluation results on QE tasks, showing the powerful transferability of UNITE model.

For future work, we believe that exploring the domain adaption problem for QE is an essential task. The existing machine translation system has made great progress in the field of domain transferablity (Lin et al., 2021; Yao et al., 2020; Wan et al., 2022b). Nevertheless, the confident evaluation metrics for those translation systems are few to be explored. Apart from that, developing a unified framework with high transferability for evaluating translation and other natural language generation tasks (Yang et al., 2021, 2022; Liu et al., 2022) is quite an interesting direction.

Notably, we also participated in this year's WMT Metrics Shared Task with the same models. We believe that, the idea of unifying three kinds of translation evaluation functionalities (*i.e.*, source-only, reference-only, and source-reference-combined) into one single model can deliver dominant results on all scenarios. Better solutions for achieving this goal are worth to be explored in the future.

Acknowledgements

The participants would like to send great thanks to the committee and the organizers of the WMT Quality Estimation Shared Task competition. Besides, the authors would like to thank the reviewers and meta-review for their insightful suggestions.

This work was supported in part the by the National Key Research and Development Program of China (No. 2020YFB1406703), Science and Technology Development Fund, Macau SAR (Grant No. 0101/2019/A2), the Multi-year Research Grant from the University of Macau (Grant No. MYRG2020-00054-FST), National Key Research and Development Program of China (No. 2018YFB1403202), and Alibaba Group through Alibaba Innovative Research (AIR) Program.

References

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 315–321, Geneva, Switzerland. COLING.

Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017. Results of the WMT17 metrics shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 489–513, Copenhagen, Denmark. Association for Computational Linguistics.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Peter Emerson. 2013. The Original Borda Count and Partial Voting. *Social Choice and Welfare*, 40(2):353–358.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021a. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021b. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Taisiya Glushkova, Chrysoula Zerva, Ricardo Rei, and André F. T. Martins. 2021. Uncertainty-aware machine translation evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3920–3938, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tin Kam Ho, Jonathan J. Hull, and Sargur N. Srihari. 1994. Decision Combination in Multiple Classifier Systems. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 16(1):66–75.

Huan Lin, Baosong Yang, Liang Yao, Dayiheng Liu, Haibo Zhang, Jun Xie, Min Zhang, and Jinsong Su. 2022. Bridging the gap between training and inference: Multi-candidate optimization for diverse neural machine translation. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2622–2632, Seattle, United States. Association for Computational Linguistics.

Huan Lin, Liang Yao, Baosong Yang, Dayiheng Liu, Haibo Zhang, Weihua Luo, Degen Huang, and Jinsong Su. 2021. Towards user-driven neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4008–4018, Online. Association for Computational Linguistics.

Xin Liu, Dayiheng Liu, Baosong Yang, Haibo Zhang, Junwei Ding, Wenqing Yao, Weihua Luo, Haiying Zhang, and Jinsong Su. 2022. Kgr4: Retrieval, retrospect, refine and rethink for commonsense generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11029–11037.

Qingsong Ma, Ondřej Bojar, and Yvette Graham. 2018. Results of the WMT18 metrics shared task: Both characters and embeddings achieve good performance. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 671–688, Belgium, Brussels. Association for Computational Linguistics.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume*

2: Shared Task Papers, Day 1), pages 62–90, Florence, Italy. Association for Computational Linguistics.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020. Results of the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Jeonghyeok Park, Hyunjoong Kim, and Hyunchang Cho. 2021. Papago's submissions to the WMT21 triangular translation task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 341–346, Online. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. TransQuest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5070–5081, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the WMT 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 684–725, Online. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Varvara Logacheva, Ramón F. Astudillo, and André F. T. Martins. 2018. Findings of the WMT 2018 shared task on quality estimation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 689–709, Belgium, Brussels. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yu Wan, Dayiheng Liu, Baosong Yang, Tianchi Bi, Haibo Zhang, Boxing Chen, Weihua Luo, Derek F. Wong, and Lidia S. Chao. 2021. RoBLEURT submission for WMT2021 metrics task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1053–1058, Online. Association for Computational Linguistics.

Yu Wan, Dayiheng Liu, Baosong Yang, Haibo Zhang, Boxing Chen, Derek Wong, and Lidia Chao. 2022a. UniTE: Unified translation evaluation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8117–8127, Dublin, Ireland. Association for Computational Linguistics.

Yu Wan, Baosong Yang, Derek F. Wong, Yikai Zhou, Lidia S. Chao, Haibo Zhang, and Boxing Chen. 2020. Self-paced learning for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1074–1080, Online. Association for Computational Linguistics.

Yu Wan, Baosong Yang, Derek Fai Wong, Lidia Sam Chao, Liang Yao, Haibo Zhang, and Boxing Chen. 2022b. Challenges of neural machine translation for short texts. *Computational Linguistics*, 48(2):321–342.

Jiayi Wang, Ke Wang, Boxing Chen, Yu Zhao, Weihua Luo, and Yuqi Zhang. 2021. QEMind: Alibaba's submission to the WMT21 quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 948–954, Online. Association for Computational Linguistics.

Xiangpeng Wei, Heng Yu, Yue Hu, Rongxiang Weng, Weihua Luo, and Rong Jin. 2022. Learning to generalize to more: Continuous semantic augmentation for neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7930–7944, Dublin, Ireland. Association for Computational Linguistics.

Kexin Yang, Wenqiang Lei, Dayiheng Liu, Weizhen Qi, and Jiancheng Lv. 2021. POS-Constrained Parallel Decoding for Non-autoregressive Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5990–6000, Online. Association for Computational Linguistics.

Kexin Yang, Dayiheng Liu, Wenqiang Lei, Baosong Yang, Haibo Zhang, Xue Zhao, Wenqing Yao, and Boxing Chen. 2022. GCPG: A general framework for controllable paraphrase generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4035–4047, Dublin, Ireland. Association for Computational Linguistics.

Liang Yao, Baosong Yang, Haibo Zhang, Boxing Chen, and Weihua Luo. 2020. Domain transfer based data augmentation for neural query translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4521–4533, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Chrysoula Zerva, Frédéric Blain, Ricardo Rei, Piyawat Lertvittayakumjorn, José G. C. de Souza, Steffen Eger, Diptesh Kanojia, Duarte Alves, Constantin Orăsan, Marina Fomicheva, André F. T. Martins, and Lucia Specia. 2022. Findings of the wmt 2022 shared task on quality estimation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Chrysoula Zerva, Daan van Stigt, Ricardo Rei, Ana C Farinha, Pedro Ramos, José G. C. de Souza, Taisiya Glushkova, Miguel Vera, Fabio Kepler, and André F. T. Martins. 2021. IST-unbabel 2021 submission for the quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 961–972, Online. Association for Computational Linguistics.

Tong Zhang, Wei Ye, Baosong Yang, Long Zhang, Xingzhang Ren, Dayiheng Liu, Jinan Sun, Shikun Zhang, Haibo Zhang, and Wen Zhao. 2022. Frequency-aware contrastive learning for neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11712–11720.

Yikai Zhou, Baosong Yang, Derek F. Wong, Yu Wan, and Lidia S. Chao. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6934–6944, Online. Association for Computational Linguistics.

KU X Upstage's submission for the WMT22 Quality Estimation: Critical Error Detection Shared Task

Sugyeong Eo[1], **Chanjun Park**[1,2], **Hyeonseok Moon**[1], **Jaehyung Seo**[1], **Heuiseok Lim**[1*]
[1]Korea University [2]Upstage
{djtnrud,bcj1210,glee889,seojae777,limhseok}@korea.ac.kr
chanjun.park@upstage.ai

Abstract

This paper presents KU X Upstage's submission to the quality estimation (QE): critical error detection (CED) shared task in WMT22. We leverage the XLM-RoBERTa large model without utilizing any additional parallel data. To the best of our knowledge, we apply prompt-based fine-tuning to the QE task for the first time. To maximize the model's language understanding capability, we reformulate the CED task to be similar to the masked language model objective, which is a pre-training strategy of the language model. We design intuitive templates and label words, and include auxiliary descriptions such as demonstration or Google Translate results in the input sequence. We further improve the performance through the template ensemble, and as a result of the shared task, our approach achieve the best performance for both English-German and Portuguese-English language pairs in an unconstrained setting.

1 Introduction

This paper presents our submission to the critical error detection (CED) shared task among the quality estimation (QE) tasks of WMT22 (Zerva et al., 2022). CED is a task of detecting cases where translation errors in source sentences or translation results distort meaning in terms of race, gender, safety, law, finance, etc. (Specia et al., 2021; Rubino et al., 2021; Jiang et al., 2021). Critical translation errors in the shared task appear in the form of mistranslation, hallucination, and deletion in source sentences or translation results, and errors can be classified into five categories: additions, deletions, named entities, meaning, and numbers. Even if machine translation (MT) systems produce fluent translations, the fact that they cannot be free from fatal semantic errors emphasizes the importance of preventing social repercussions from the errors. Forbidding socially bad influences and losses from these meaning deviations is the purpose of the CED task (Specia et al., 2021).

Participating systems distinguish only critical errors, not correct translations or simple translation errors. In contrast to last year, submissions should be provided with continuous scores rather than binary labels. The official script calculates scores with automatically assigned classes based on the threshold value of the index corresponding to the number of errors. Similar to last year, we participated in unconstrained English-German (En-De) and Portuguese-English (Pt-En) utilizing released training datasets[1].

To perform the CED task, we exploit the XLM-RoBERTa large model (Conneau et al., 2019) as utilized in the baseline without additional parallel data. In addition, we adopt prompt-based fine-tuning to mitigate catastrophic forgetting during fine-tuning by maximizing the linguistic capability obtained through pre-training. In prompt-based fine-tuning, the downstream task is reformulated into a cloze-style, which is consistent with the masked language modeling objective. The word for the masked part is predicted by the model based on the task-specific template (Liu et al., 2021a). Recent studies have demonstrated the remarkable effects of prompt-based learning in the natural language processing field (Brown et al., 2020; Gao et al., 2020; Schick and Schütze, 2020; Liu et al., 2021b; Zhao and Schütze, 2021), and we apply this new paradigm to the QE task. We manually generate templates each containing a source sentence, its translation result, and a description with a mask token for the CED task. Furthermore, we generate label words (Liu et al., 2021b) to map the words to be filled in the masked part and labels.

Exploring appropriate templates in prompt-based fine-tuning is important because the performance ranges widely depending on the template

[*] Corresponding Author

[1] The following is the leaderboard of the CED task. https://codalab.lisn.upsaclay.fr/competitions/6893

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 606 - 614
December 7-8, 2022 ©2022 Association for Computational Linguistics

used. Therefore, we design multiple hard prompts through prompt engineering, and these are configured into three types of templates according to additional information: plain template, template with demonstration, and template with Google Translate. Through answer engineering, we map contrastive words for each OK and BAD tag in diverse combinations. To obtain the final score, we extract probability for words mapped to BAD. We further improve performance by ensembling values from templates.

Our approach outperforms the baseline models in En-De and Pt-En by a substantial margin and achieves first place. Experimental results demonstrate that simply setting up the training method without modifying the model or augmenting the data with additional parallel corpora significantly affects the performance.

Template
<s> **src** </s> **mt**. <mask> translation.</s>
<s> **src** </s> **mt**. It was <mask> translation.</s>
<s> A <mask> translation of **src** is **mt**.</s>
<s> **src** </s> **mt** <mask></s>
<s> **src** </s> **mt**? <mask></s>
<s> **src** </s> **mt**? <mask>,</s>
<s> **src** </s> **mt**? "<mask>"</s>

Label Words
OK: "great", BAD: "terrible"
OK: "good", BAD: "bad"
OK: "!", BAD: "?"
OK: "nice", BAD: "poor"
OK: "yes", BAD: "no"

Table 1: Prompt templates and label words utilized in our experiments. We denote a source sentence as **src** and its translation result as **mt**.

2 Proposed Method

2.1 Prompt-based Fine-tuning

We adopt prompt-based fine-tuning to diminish the discrepancy between the training objectives of the fine-tuning and pre-training (Shin et al., 2020). By applying this, we induce our CED model to preserve the linguistic capability obtained via the pre-training phase.

In our task, we denote $(src, mt, y) \in D$ for a CED training dataset D, where src and mt denote a source sentence and its translated sentence, respectively, and y denotes its corresponding label (*e.g.* OK, BAD). Furthermore, we define two mapping functions T, L that transform all the data in D to implement prompt-based fine-tuning in the CED task.

The template function T transforms each src and mt into a single input sequence that contains description with masked token. In generating the input sequence, T also defines the placement of a special <mask> token to fill in. During training, we induce the model to infer the appropriate word suitable for the corresponding <mask> token position that is coherent with the overall context. Subsequently, the label word function, referred to as verbalizer, L transforms the given label y into an appropriate label word to be placed in the masked position of the input sequence transformed through T.

For example, given src as "indigenous peoples constitute just 0.7% of the global population", mt as "Indigene Völker machen nur 5% der Weltbevölkerung aus", and their corresponding label y as BAD with label words "OK:great, BAD:terrible", T convert these sentences into "<s> indigenous peoples constitute just 0.7% of the global population </s> Indigene Völker machen nur 5% der Weltbevölkerung aus </s>. It was <mask> translation." and L convert its label into "terrible". Then the original fine-tuning objective of CED that determines whether the label is "OK" or "BAD" is converted to predict the correct word for the <mask> token position. Specifically, the model is trained to predict the following probability:

$$P(y|src, mt) = P(\langle mask \rangle = L(y)|T(src, mt)) \quad (1)$$

Considering the scoring method of the WMT22 CED task, we do not binarize the model inference results into a OK or a BAD tag. Instead, we use the softmax function to normalize the overall score as in Equation (2) and extract the probability that the decoded token in the <mask> position will be mapped to BAD. We regard this probability as the estimated quality score of the mt.

$$\text{score}(src, mt) = \frac{\exp(P(\texttt{BAD}|src, mt))}{\sum_{y' \in \{\texttt{OK},\texttt{BAD}\}} \exp(P(y'|src, mt))} \quad (2)$$

2.2 Prompt and Answer Engineering

Because the effective prompt for the CED task has not been revealed, we design various prompt candidates (Gao et al., 2020). We attempt to organize the model input into a natural context, such as "src

	En-De			Pt-En		
	Train	Dev	Test	Train	Dev	Test
# of Sentences	155511	17280	500	39925	4437	500
Avg **`src`** Toks	22.98	23.07	24.15	25.49	25.5	26.63
Avg **`mt`** Toks	23.71	23.8	24.68	22.52	22.39	23.26
Min/Max **`src`** Toks	2/112	2/90	4/82	2/117	2/85	3/74
Min/Max **`mt`** Toks	2/106	2/109	4/80	1/107	2/82	3/69
% of **BAD** label	6.1	5.82	-	6.05	5.79	-

Table 2: Dataset statistics on WMT22 CED task

mt. It was <mask> translation, A <mask> translation of src is mt". For the label words, we select two distinct words, such as "great/terrible", and "good/bad". We intend to obviate ambiguity during model training by establishing clear contrasting label words, although naive errors are not considered a good translation result. All types of templates and label words are listed in Table 1, and the entire prompt used in our experiments is described in Appendix A.

2.3 Auxiliary Description

We append auxiliary descriptions that provide supplementary information to the model input (Gao et al., 2020; Chen et al., 2021; Brown et al., 2020). We select two types of auxiliary descriptions: demonstration and Google Translate results.

The demonstration extracts a single example for each class from the training data and concatenates them into the input sequence, similar to the in-context learning approach proposed in GPT3 (Brown et al., 2020) and LM-BFF (Gao et al., 2020). In contrast to LM-BFF, we randomly select training examples without any constraints on sampling to avoid unintended bias that may occur when extracting demonstrations based on semantic similarity.

The Google Translate results append translation results from the commercialized MT system. As demonstrated in previous studies (Chen et al., 2021; Wang et al., 2020; Moon et al., 2021), adding Google Translate results contributes to a significant performance improvement. Regarding this, we use Google Translate to generate mt' by translating each src in the entire data. By adding this to the input sequence, we distill the knowledge of the external MT system into the model.

Auxiliary descriptions are combined with each example in D to compose a new input sequence. Through this, we induce the model to determine the critical errors by grounding more information.

2.4 Prompt Ensemble

As mentioned previously, prompt-based fine-tuning shows various deviations in model performance depending on the designed prompts (Shin et al., 2020). We aim to boost performance by aggregating the results from multiple prompts to minimize bias and distribute contributions per template. For the ensemble, we add the top K values with high Matthew's correlation coefficient (MCC) results.

3 Experimental Setting

3.1 Dataset Details

We leverage the dataset provided by WMT22[2]. The dataset statistics for each language pair are reported in Table 2. In summary, a sentence contains an average of 22 to 26 tokens, with a bad tag ratio of 5-6%. When using auxiliary descriptions, we randomly extract data corresponding to `OK` and `BAD` tags from the training dataset to configure the demonstration. When leveraging the commercialized MT result, we translate source sentences using the most widely adopted Google Translate[3].

We tokenize sentences with the XLM-RoBERTa tokenizer. Considering the average token and maximum sequence length of statistics, after concatenating src and mt, we filter cases where the tokenized sentence length is over 250. We score our predictions with the official script[4] provided by WMT22 and MCC.

3.2 Model Details

We exploit the same multilingual language model, XLM-RoBERTa large (Conneau et al., 2019), for both En-De and Pt-En language pairs and leverage the model and tokenizer[5] distributed by Huggingface (Wolf et al., 2019). For conducting prompt-based fine-tuning, we experiment after modifying LM-BFF[6] framework. In the case of hyperparameters, the max sequence length is set to 256 and batch size is set to 32 if auxiliary description is not used in model training, otherwise we set the max sequence length to 350 and batch size to 16. As shown in Table 2, considering the total data size for each language pair, we train Pt-En to 10K training

[2] `https://github.com/WMT-QE-Task/wmt-qe-2022-data`

[3] `https://translate.google.co.kr/`

[4] `https://github.com/WMT-QE-Task/wmt-qe-2022-data/blob/main/critical-errors-subtask/official_evaluation.py`

[5] `xlm-roberta-large`

[6] `https://github.com/princeton-nlp/LM-BFF.git`

	En-De			Pt-En		
	MCC (Binary)	MCC	P&R	MCC (Binary)	MCC	P&R
Baseline	-	0.8943	0.9001	-	0.8955	0.9012
Plain (Avg)	**0.9161 ±0.0037**	**0.9117 ±0.0075**	**0.9166 ±0.0071**	0.9223 ±0.0089	0.9042 ±0.0217	0.9095 ±0.0206
Demo (Avg)	0.9121 ±0.0062	0.9072 ±0.0115	0.9123 ±0.0109	0.9118 ±0.0113	0.9003 ±0.0266	0.9053 ±0.0246
Google MT (Avg)	0.9143 ±0.0272	0.9092 ±0.0217	0.9142 ±0.0205	**0.9391 ±0.0331**	**0.9312 ±0.0444**	**0.9350 ±0.0238**
Plain (Max)	0.9189	0.9153	0.9200	0.9312	0.9173	0.9218
Demo (Max)	0.9183	**0.9187**	0.9160	0.9231	0.9173	0.9177
Google MT (Max)	**0.9218**	0.9165	**0.9211**	**0.9649**	**0.9565**	**0.9588**

Table 3: Unconstrained En-De, Pt-En development (dev) set result on WMT22 CED task. We measure MCC from the WMT22 official script. As the official script refines the inference results to have the same distribution of OK and BAD with reference, precision and recall always indicate the same value. Therefore, we denote precision and recall as P&R. We further present the MCC (Binary) result measured through the binary label. This result tends to be higher than the official script.

	En-De		Pt-En	
	MCC	P&R	MCC	P&R
All	0.9265	0.9305	0.9565	0.9588
Top 5	0.9309	0.9347	0.9695	0.9712
Top 10	0.9309	0.9347	**0.9739**	**0.9753**
Top 15	**0.9321**	**0.9358**	0.9652	0.9671
Truncate	0.9287	0.9326	0.9521	0.9547

Table 4: MCC results on top K template ensemble. Truncate indicates an ensemble result only when the dev MCC is over the baseline result.

steps and En-De to 35K. As a GPU setting, one RTX 8000 is used for learning.

4 Experimental Results

4.1 Prompt-based Fine-tuning Results

We present the prompt-based fine-tuning results for En-De and Pt-En language pairs in Table 3. We mainly divide results into three categories: plain template, template with demo, and template with Google Translate according to the auxiliary description we used. Each consists of 8, 11, and 20 different templates, and we report the average and max values in the table. Performances by leveraging each template is described in Appendix A. The baseline is the official fine-tuning results for the XLM-RoBERTa large model. Our approach significantly outperforms the baseline performance in average and maximum performance for all experiments.

Specifically, templates with no auxiliary description (*i.e.* Plain) show comparatively high results in En-De. When using demonstration (*i.e.* Demo) and Google Translate (*i.e.* Google MT), the performance is slightly decreased. However, templates with a demonstration show effective benefits in the max MCC. In addition, the best performance is achieved in templates with Google Translate in the case of MCC (Binary), which measured MCC by comparing binary predictions and labels. Through the results, we conclude that including additional information in the input sequence leads to performance improvement.

Pt-En Google MT MCC results strongly support our hypothesis. Additional translation results within the input sequence competitively contribute to performance improvement in both average and max, outperforming +0.0392 MCC over the Plain (Max). When comparing demonstration and Google Translate, we infer that presenting information related to the input example has a better effect on learning than providing representative examples of tasks.

In the average results (*i.e.* Avg), the performance gap per template is indicated by ±. Under the setting where the selected auxiliary description is fixed, the performance of different templates varies considerably, from 0.0037 to 0.0217 MCC for En-De and from 0.0089 to 0.444 MCC for Pt-En. Therefore, we perform ensembles to obtain the final score by aggregating the top K predictions.

4.2 Template Ensemble Results

Table 4 is the ensemble results of the top K templates, showing notable performance. Ensembles against the top 15 templates for En-De and the top 10 templates for Pt-En yield the best MCC results. These show +0.0134 MCC higher in En-De and +0.0174 MCC higher in Pt-En than the max results listed in Table 3. This demonstrates that the distributed contribution to multiple prompts per example further improves the final performance.

	En-De		Pt-En	
	MCC	P&R	MCC	P&R
Baseline	0.855	0.873	0.934	0.944
aiXplain	0.219	0.318	0.179	0.296
Ours	**0.964**	**0.968**	**0.984**	**0.986**

Table 5: Official result on En-De, Pt-En CED blind test set

Furthermore, we note All and Truncate in the table. The former ensembles all results and the latter removes templates with lower results than the baseline evaluation MCC before ensembling. Through this, we observe that including most of the templates does not necessarily contribute to performance improvement. High performance is obtained by training models with various types of templates and selecting appropriate predictions together.

4.3 Results on Test dataset

The experimental results for the test set are shown in Table 5. We submit the final score obtained through the ensemble. As a result, we significantly outperform the baseline result, achieving +0.109 MCC in En-De and +0.05 MCC in Pt-En. This is a notable margin because we use the same model as the baseline without utilizing any supplementary parallel data or scaling model parameters.

5 Conclusion

We applied prompt-based learning to the CED task by forming a learning objective for the task similar to that in pre-training. This method outperformed the baseline performance while preserving the model parameters and data settings. We performed manual prompt engineering and answer engineering to explore intuitive hard prompts. In addition, because finding optimal prompts is difficult, we ensembled predictions from diverse templates to address the performance variation and achieve additional performance boost. Our method is simple but powerful, and we hope that this method will be actively introduced to QE tasks in future studies.

Limitations

This study used models trained only on English-German and Portuguese-English language pairs. Therefore, language extension is challenging because data for training the CED task must be prepared for each language pair and direction. Non-trivial costs may be incurred in the data construction process. Furthermore, because we manually generated prompts and answer engineering, finding the optimal prompt is sub-optimal. Soft prompts that leverage the trained embedding values in the prompt configuration can mitigate this limitation. However, because embedding vectors in soft prompt are not described in human words, and we are the first to introduce prompt-based learning in QE tasks, we focused on interpretability.

Ethics Statement

We created task-specific templates during prompt engineering. We have not used problematic statements at this time. In addition, when engineering label words, words without ethical issues were used. However, unethical expressions such as socially problematic words and abusive language are included in the CED data. Owing to the nature of the task, this is intentionally appended by annotators to detect critical errors. The purpose of this task is to classify and exclude ethical issues that occur in the machine translation field.

Acknowledgements

This research was supported by the Ministry of Science and ICT (MSIT), Korea, under the Information Technology Research Center (ITRC) support program (IITP-2022-2018-0-01405) supervised by the Institute for Information & Communications Technology Planning & Evaluation(IITP) and this work was supported by an IITP grant funded by the MSIT (No. 2020-0-00368, A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques) and this research was supported by the MSIT, Korea, under the ICT Creative Consilience program (IITP-2022-2020-0-01819) supervised by the IITP.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yimeng Chen, Chang Su, Yingtao Zhang, Yuxia Wang, Xiang Geng, Hao Yang, Shimin Tao, Guo Jiaxin, Wang Minghan, Min Zhang, et al. 2021. Hw-tsc's participation at wmt 2021 quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 890–896.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.

Genze Jiang, Zhenhao Li, and Lucia Specia. 2021. Icl's submission to the wmt21 critical error detection shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 928–934.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pretrain, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

Hyeonseok Moon, Chanjun Park, Sugyeong Eo, Jaehyung Seo, and Heuiseok Lim. 2021. An empirical study on automatic post editing for neural machine translation. *IEEE Access*, 9:123754–123763.

Raphaël Rubino, Atsushi Fujita, and Benjamin Marie. 2021. Nict kyoto submission for the wmt'21 quality estimation task: Multimetric multilingual pretraining for critical error detection. In *Proceedings of the Sixth Conference on Machine Translation*, pages 941–947.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André FT Martins. 2021. Findings of the wmt 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 684–725.

Minghan Wang, Hao Yang, Hengchao Shang, Daimeng Wei, Jiaxin Guo, Lizhi Lei, Ying Qin, Shimin Tao, Shiliang Sun, Yimeng Chen, et al. 2020. Hw-tsc's participation at wmt 2020 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1056–1061.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Chrysoula Zerva, Frédéric Blain, Ricardo Rei, Piyawat Lertvittayakumjorn, José G. C. de Souza, Steffen Eger, Diptesh Kanojia, Duarte Alves, Constantin Orăsan, Marina Fomicheva, André F. T. Martins, and Lucia Specia. 2022. Findings of the wmt 2022 shared task on quality estimation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Mengjie Zhao and Hinrich Schütze. 2021. Discrete and soft prompting for multilingual models. *arXiv preprint arXiv:2109.03630*.

A Appendix

A.1 Results on Each Template

The evaluation MCC results for each template of the En-De and Pt-En pairs are shown in Tables 6 and 7. Particularly in Pt-En, the performance of each template varies considerably, reporting significantly lower or superior performance than the baseline result.

Type	Index	Template	Label Words	MCC (Binary)	MCC	P&R
Plain Template	1	**src** **mt**. <mask>translation.	great / terrible	0.9156	**0.9153**	0.9200
	2	**src** **mt**. <mask>translation.	good / bad	0.9158	0.9087	0.9137
	3	**src** **mt** <mask>	! / ?	0.9153	**0.9131**	0.9179
	4	A <mask>translation of **src** is **mt**.	good / bad	0.9189	**0.9142**	0.9189
	5	A <mask>translation of **src** is **mt**.	great / terrible	0.9161	**0.9131**	0.9179
	6	**src** **mt**. It was <mask>translation.	great / terrible	0.9124	0.9042	0.9095
	7	**src** **mt**. It was <mask>translation.	nice / poor	0.9161	**0.9131**	0.9179
	8	**src** **mt**? <mask>	yes / no	0.9184	***0.9120***	0.9168
Template with Demo	1	**demo_ok** **demo_bad** **srcmt**. <mask>translation.	great / terrible	0.9149	0.9064	0.9116
	2	**demo_ok** **demo_bad** **src** **mt**. <mask>translation.	good / bad	0.9089	0.9098	0.9147
	3	**demo_ok** **demo_bad** **src** **mt**. It was <mask>translation.	great / terrible	0.9125	0.9064	0.9116
	4	**demo_ok** **demo_bad** **src** **mt**. It was <mask>translation.	nice / poor	0.9183	**0.9187**	0.9232
	5	**demo_ok** **demo_bad** **src** **mt**. It was <mask>translation.	! / ?	0.9084	0.9042	0.9095
	6	**demo_ok** **demo_bad** **src** **mt**? <mask>	yes / no	0.9109	0.9075	0.9126
	7	**src** **mt** **demo_ok** **demo_bad** <mask>translation.	great / terrible	0.9095	0.8964	0.9021
	8	**src** **mt** **demo_ok** **demo_bad** <mask>translation.	good / bad	0.9060	0.9042	0.9095
	9	**src** **mt** **demo_ok** **demo_bad** . It was <mask>translation.	great / terrible	0.9123	0.9064	0.9116
	10	**src** **mt** **demo_ok** **demo_bad** . It was <mask>translation.	nice / poor	0.9138	0.9098	0.9147
	11	**src** **mt** **demo_ok** **demo_bad** ? <mask>	yes / no	0.9175	0.9098	0.9147
Template with Google Translate	1	**src** **mt**? <mask>**gmt**	great / terrible	0.9161	0.9098	0.9147
	2	**src** **mt**? <mask>**gmt**	good / bad	0.9198	**0.9165**	0.9211
	3	**src** **mt**? <mask>**gmt**	! / ?	0.9218	**0.9165**	0.9211
	4	**src** **mt**? <mask>**gmt**	yes / no	0.9121	0.9087	0.9137
	5	**src** **mt**? It was <mask>. **gmt**	great / terrible	0.9173	0.9053	0.9105
	6	**src** **mt**? It was <mask>. **gmt**	good / bad	0.9166	0.9087	0.9137
	7	**src** **mt**? It was <mask>. **gmt**	! / ?	0.9172	**0.9153**	0.9200
	8	**src** **mt**? It was <mask>. **gmt**	yes / no	0.9158	***0.9120***	0.9168
	9	**src** **mt**? "<mask>", **gmt**	! / ?	0.9176	***0.9120***	0.9168
	10	**src** **mt**? "<mask>", **gmt**	good / bad	0.9175	0.9064	0.9116
	11	**src** **mt**? <mask>, **gmt**	! / ?	0.9103	0.9087	0.9137
	12	**src** **mt**? <mask>, **gmt**	good / bad	0.9111	0.9087	0.9137
	13	**src** **mt** **gmt**. <mask>translation.	great / terrible	0.9133	0.9009	0.9063
	14	**src** **mt** **gmt**. <mask>translation.	good / bad	0.8872	~~**0.8875**~~	0.8937
	15	**src** **mt** **gmt**. <mask>	! / ?	0.9151	0.9064	0.9116
	16	A <mask>translation of **src** is **mt** **gmt**.	good / bad	0.9209	**0.9165**	0.9211
	17	A <mask>translation of **src** is **mt** **gmt**.	great / terrible	0.9134	***0.9109***	0.9158
	18	**src** **mt** **gmt**. It was <mask>translation.	great / terrible	0.9134	0.9075	0.9126
	19	**src** **mt** **gmt**. It was <mask>translation.	nice / poor	0.9160	**0.9165**	0.9211
	20	**src** **mt** **gmt**? <mask>	yes / no	0.9147	0.9098	0.9147

Table 6: En-De results for all templates. The top five MCCs are in red bold, the top 10 MCCs are in orange bold and underlined, and the top 15 MCCs are in blue bold and italic. We indicate the MCC below the baseline as gray bold and strikeouts.

Type	Index	Template	Label Words	MCC (Binary)	MCC	P&R
Plain Template	1	**src mt**. <mask>translation.	great / terrible	0.9312	0.9129	0.9177
	2	**src mt**. <mask>translation.	good / bad	0.9203	0.9173	0.9218
	3	**src mt** <mask>	! / ?	0.9190	~~**0.8825**~~	0.8889
	4	A <mask>translation of **src** is **mt**.	good / bad	0.9250	0.9129	0.9177
	5	A <mask>translation of **src** is **mt**.	great / terrible	0.9246	0.9129	0.9177
	6	**src mt**. It was <mask>translation.	great / terrible	0.9170	~~**0.8868**~~	0.8930
	7	**src mt**. It was <mask>translation.	nice / poor	0.9264	0.9129	0.9177
	8	**src mt**? <mask>	yes / no	0.9150	~~**0.8955**~~	0.9012
Template with Demo	1	**demo_ok demo_bad srcmt**. <mask>translation.	great / terrible	0.9080	~~**0.8825**~~	0.8889
	2	**demo_ok demo_bad src mt**. <mask>translation.	good / bad	0.9201	0.8999	0.9053
	3	**demo_ok demo_bad src mt**. It was <mask>translation.	great / terrible	0.9178	0.9042	0.9095
	4	**demo_ok demo_bad src mt**. It was <mask>translation.	nice / poor	0.9112	0.9042	0.9095
	5	**demo_ok demo_bad src mt**. It was <mask>translation.	! / ?	0.9009	0.8999	0.9053
	6	**demo_ok demo_bad src mt**? <mask>	yes / no	0.9044	0.9129	0.9177
	7	**src mt demo_ok demo_bad** <mask>translation.	great / terrible	0.9131	0.9042	0.9095
	8	**src mt demo_ok demo_bad** <mask>translation.	good / bad	0.9056	~~**0.8737**~~	0.8807
	9	**src mt demo_ok demo_bad** . It was <mask>translation.	great / terrible	0.9199	0.9086	0.9136
	10	**src mt demo_ok demo_bad** . It was <mask>translation.	nice / poor	0.9231	***0.9173***	0.9173
	11	**src mt demo_ok demo_bad** ? <mask>	yes / no	0.9056	~~**0.8955**~~	0.9012
Template with Google Translate	1	**src mt**? <mask>**gmt**	great / terrible	0.9471	***0.9390***	0.9424
	2	**src mt**? <mask>**gmt**	good / bad	0.9558	**0.9478**	0.9506
	3	**src mt**? <mask>**gmt**	! / ?	0.9537	**0.9434**	0.9465
	4	**src mt**? <mask>**gmt**	yes / no	0.9580	**0.9565**	0.9588
	5	**src mt**? It was <mask>. **gmt**	great / terrible	0.9515	**0.9521**	0.9547
	6	**src mt**? It was <mask>. **gmt**	good / bad	0.9649	**0.9565**	0.9588
	7	**src mt**? It was <mask>. **gmt**	! / ?	0.9470	**0.9521**	0.9547
	8	**src mt**? It was <mask>. **gmt**	yes / no	0.9625	**0.9521**	0.9547
	9	**src mt**? "<mask>", **gmt**	! / ?	0.9430	***0.9390***	0.9424
	10	**src mt**? "<mask>", **gmt**	good / bad	0.9603	**0.9521**	0.9547
	11	**src mt**? <mask>, **gmt**	! / ?	0.9538	**0.9521**	0.9547
	12	**src mt**? <mask>, **gmt**	good / bad	0.9514	**0.9478**	0.9506
	13	**src mt gmt**. <mask>translation.	great / terrible	0.9252	***0.9173***	0.9218
	14	**src mt gmt**. <mask>translation.	good / bad	0.9219	0.9086	0.9136
	15	**src mt gmt**. <mask>	! / ?	0.9269	***0.9173***	0.9218
	16	A <mask>translation of **src** is **mt gmt**.	good / bad	0.9060	~~**0.8912**~~	0.8971
	17	A <mask>translation of **src** is **mt gmt**.	great / terrible	0.9125	~~**0.8868**~~	0.8930
	18	**src mt gmt**. It was <mask>translation.	great / terrible	0.9073	0.9042	0.9095
	19	**src mt gmt**. It was <mask>translation.	nice / poor	0.9185	0.9086	0.9136
	20	**src mt gmt**? <mask>	yes / no	0.9147	0.8999	0.9053

Table 7: Pt-En results on all templates. The color and the style of top K performances are equivalent to Table 6.

NJUNLP's Participation for the WMT2022 Quality Estimation Shared Task

Xiang Geng[1], **Yu Zhang**[1], **Shujian Huang**[1]*, **Shimin Tao**[2], **Hao Yang**[2], **Jiajun Chen**[1]
[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[2] Huawei Translation Services Center, Beijing, China
{gx, zhangy}@smail.nju.edu.cn, {huangsj,chenjj}@nju.edu.cn
{taoshimin, yanghao30}@huawei.com

Abstract

This paper presents submissions of the NJUNLP team in WMT 2022 Quality Estimation shared task 1, where the goal is to predict the sentence-level and word-level quality for target machine translations. Our system explores pseudo data and multi-task learning. We propose several novel methods to generate pseudo data for different annotations using the conditional masked language model and the neural machine translation model. The proposed methods control the decoding process to generate more real pseudo translations. We pre-train the XLMR-large model with pseudo data and then fine-tune this model with real data both in the way of multi-task learning. We jointly learn sentence-level scores (with regression and rank tasks) and word-level tags (with a sequence tagging task). Our system obtains competitive results on different language pairs and ranks first place on both sentence- and word-level sub-tasks of the English-German language pair.

1 Introduction

Quality Estimation (QE) of Machine Translation (MT) is a task to predict the quality of translations at run-time without relying on reference translations (Specia et al., 2018). This paper describes the contribution of the NJUNLP team to the WMT2022 QE Shared Task (Zerva et al., 2022) on sentence- and word-level sub-tasks (task 1)[1]. For the sentence-level task, participating systems are required to predict the quality score for each translation output, and all scores are standardized using the z-score by the rater. The result is evaluated using Spearman's rank correlation coefficient as the primary metric. For the word-level task, participating systems are required to tag each token of the translation output with OK and BAD. The BAD tag denotes this token is wrong, or there is one or more missing token(s) on the left side. The result is evaluated in terms of Matthews correlation coefficient (MCC) as the primary metric.

Inspired by DirectQE(Cui et al., 2021), we further explore pseudo data and multi-task learning for the QE shared task. Our main contributions are as follows:

- We propose several novel methods to generate pseudo data for different annotations using the conditional masked language model (Cui et al., 2021) and the neural machine translation model (Vaswani et al., 2017).
- We use the XLMR-large model (Conneau et al., 2020) as the QE model rather than a transformer base model with random initialization in (Cui et al., 2021).
- We pre-train the QE model with pseudo data and then fine-tune it with real data both in the way of multi-task learning. We explore the rank task in addition to commonly used regression and sequence tagging tasks.
- We also explore post-editing annotation data of the previous years for the multi-dimensional quality metrics (MQM) annotation sub-task.
- We propose a new ensemble technique for combining the scores of models trained with different sentence-level scores.

Our system obtains competitive results on different language pairs. Moreover, we rank first place on both sentence- and word-level of the English-German language pair with the Spearman score of 63.47 (+1.33 than the second best system) and MCC score of 35.19 (+3.33).

* Corresponding Author.

[1] https://wmt-qe-task.github.io/subtasks/task1/

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 615 - 620
December 7-8, 2022 ©2022 Association for Computational Linguistics

Source	The light from the Earth, some of it falls in, but some of it gets lensed around and brought back to us.		
Translation	Das Licht von der Erde, einiges davon fällt hinein, aber einiges davon wird herumlinsiert und zu uns zurückgebracht.		
Annotation ID	**Error Span**	**Category**	**Severity**
Span 1	*einiges davon fällt hinein, aber einiges davon*	Style/Awkward	Major
Span 2	*herumlinsiert und zu uns zurückgebracht*	Accuracy/Mistranslation	Major
MQM	0.4444		

Table 1: An example from the WMT2022 English-German MQM dataset. We mark the error span with an italic font.

2 Sentence- and Word-Level Task

Formally, given a source language sentence $\mathbf{X}$ and a target language translation $\hat{\mathbf{Y}} = \{y_1, y_2, \ldots, y_n\}$ with n tokens. The sentence-level score m denotes the whole quality of the target $\hat{\mathbf{Y}}$. The word-level labels is a sequence of n tags $\mathbf{G} = \{g_1, g_2, \ldots, g_n\}$. g_j is the quality label for the word translation y_j, which is a binary label (OK or BAD).

In WMT2022, sentence scores are derived not only using direct assessments (DA) (Graham et al., 2013; Guzmán et al., 2019; Fomicheva et al., 2020) but also multi-dimensional quality metrics (MQM) (Burchardt and Lommel, 2014; Freitag et al., 2021). Similarly, organizers derive word tags in two different ways: Post-Editing (PE) (Snover et al., 2006; Fomicheva et al., 2020) and MQM. Moreover, MQM is introduced for the first time in the sentence- and word-Level QE shared task. MQM provides fine-grained error annotations produced by human translators. Annotators are instructed to span all errors in translation $\hat{\mathbf{Y}}$ given source sentence $\mathbf{X}$. Besides, they annotate categories and severity levels (minor, major, and critical) for these errors. According to the number of errors at different severity levels, the MQM score can be calculated as follows:

$$\mathrm{MQM} = 1 - \frac{n_{\text{minor}} + 5n_{\text{major}} + 10n_{\text{critical}}}{n}. \quad (1)$$

We show an example in Table 1.

3 Methods

To handle the few-shot and zero-shot settings, we follow DirectQE (Cui et al., 2021) framework. Specifically, we first generate pseudo data using parallel data, then pre-train the QE model with generated data, and fine-tune the pre-trained model with real QE data if provided. We will describe these steps as follows.

3.1 Pseudo Data

3.1.1 MQM Annotations

DirectQE randomly replaces some target tokens in parallel pairs with tokens sampled from the conditional masked language model. The replaced tokens are annotated as BAD, and they denote the ratio of BAD tokens as the pseudo sentence scores. There are several gaps between DirectQE pseudo data and MQM data:

- **Error distribution:** DirectQE generates errors at the token-level while MQM annotates translations with spans.
- **Error severity levels:** DirectQE uses the same sampling strategy and assigns the same weight for every pseudo error. As mentioned above, MQM assigns different weights for errors with varying levels of severity.
- **Error categories:** DirectQE does not involve error types of over- and under-translations, which are essential in real applications.
- **Generator:** DirectQE only uses a conditional masked language model as the generator for pseudo translations. This generator could perform quite differently from the target machine translation system.

To handle these problems, we proposed two novel methods to generate pseudo MQM data with different generators: the conditional masked language model and the neural machine translation model. The conditional masked language model is trained using masked language model task (Devlin et al., 2019) conditioned on the source sentence. Please refer DirectQE (Cui et al., 2021) for more details. The neural machine translation model is a common transformer base model as described in (Vaswani et al., 2017).

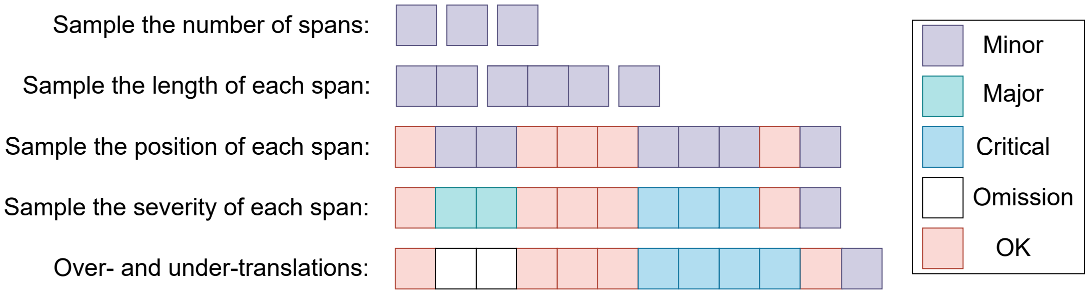

Sample the number of spans:
Sample the length of each span:
Sample the position of each span:
Sample the severity of each span:
Over- and under-translations:
Minor
Major
Critical
Omission
OK

Figure 1: Illustration of the proposed method for generating pseudo MQM data. Given a reference sentence with eleven OK tokens, we randomly sample three error spans with a length of two, three, and one and the severity of major, critical, and minor. Besides, we randomly insert one token in the second span and remove all tokens from the first span to simulate over- and under-translations.

To simulate the target error distribution, we first count the number of spans in each translation, the length of each span, and the frequency of different severity levels. Then, we can sample pseudo errors according to the target error distribution as shown in Figure 1. Finally, we use the generator to generate these error tokens except for omissions. The conditional masked language model generates pseudo errors parallel, while the neural machine translation model generates these errors from left to the right in an autoregressive fashion. Similar to DirectQE, we random sample one of the tokens with the top k generation probability as the error token. We use bigger k for graver pseudo errors to simulate errors at different severity levels. Empirically, we set k as 2, 10, and 100 for minor, major, and critical errors, respectively. The pseudo MQM scores can be calculated according Eq. 1.

3.1.2 DA and PE Annotations

For DA and PE annotations, we also explore the above two generators with different generation processes. We use the conditional masked language model as described in DirectQE. The only difference is that we normalize the pseudo sentence scores using the z-score because these scores are on a different scale from real scores.

We utilize the neural machine translation model in quite a different way. Instead of replacing target tokens at random, we let the neural machine translation model decide which tokens need to be replaced. Specifically, we compare the generation probability $P_i = \log P(y_i|X, y_{<i}; \theta_{\mathrm{MT}})$ of i-th reference token with ϵ. If $P_i < \epsilon$, we replace y_i with $y_{\max} = \arg\max_y \log P(y|X, y_{<i}; \theta_{\mathrm{MT}})$ whose generation probability is highest at this position and tag this token as BAD. Empirically, we set ϵ according to the different corpus. In addition, whatever the generation probability is, we have a chance of forcing the generated token to be consistent with the reference one. In this way, we can avoid the phenomenon that the generation probabilities of the reference token are always on a low level because of continuous replacement.

3.2 Pre-training and Fine-tuning

3.2.1 QE Model

Recently, many QE works have focused on transferring knowledge from large pre-trained language models for the QE task. In this study, we adopt XLMR large model (Conneau et al., 2020) as our QE model instead of a transformer base model with random initialization as described in (Cui et al., 2021). The XLMR large model, successfully used in the QE task(Ranasinghe et al., 2020), is a cross lingual pre-trained sentence encoder. Thus, we concatenate both source and target sentences as the input. We directly use the corresponding outputs from the last layer as token representations. We average sub-tokens' representations as the representation of the whole word. We average the representations of all target tokens as the score representation. We use linear layers for predicting sentence scores and word tags with these representations.

3.2.2 Multi-task Learning

Multi-task learning has been widely studied for QE task (Fan et al., 2019; Cui et al., 2021). Usually, the word-level task is formulated as a sequence labeling problem using cross-entropy (CE) loss as follows:

$$L_{\mathrm{CE}} = \sum_{i=1}^{n} \mathrm{CE}(g_i, \hat{g}_i), \tag{2}$$

Annotation	Pair	Spearman (Rank)	MCC (Rank)	F1-BAD	F1-OK
MQM	EN-DE	63.47 (1)	35.19 (1)	35.09	98.03
	EN-RU	47.42 (4)	38.98 (3)	43.96	94.90
	EN-ZH	29.56 (7)	30.84 (3)	30.25	98.77
	Multilingual	46.82 (2)	-	-	-
PE and DA	EN-MR	58.47 (4)	41.16 (2)	47.22	93.86
	KM-EN	-	42.12 (3)	74.42	67.68

Table 2: Results on different test sets of WMT2022.

where $\hat{g}_i$ denotes the tag predicted for i-th word. Traditional methods formulate the sentence-level task as a constraint regression problem with mean square error (MSE) loss:

$$L_{\text{MSE}} = \text{MSE}(m, \hat{m}), \tag{3}$$

where $\hat{m}$ denotes the output score. However, the ordinal relations between different translations are more important in many real applications, such as re-ranking for candidate translations and selecting the best translation models. Therefore, we introduce the additional rank loss to model the ordinal information between translations:

$$L_{\text{Rank}} = \max(0, -r(\hat{m}^i - \hat{m}^j) + \epsilon), \tag{4}$$

where $\hat{m}^i$ and $\hat{m}^j$ denote the output scores of i-th and j-th translations from current batch; r denotes the rank label, $r = 1$ if $m^i > m^j$, $r = -1$ if $m^i < m^j$; ϵ denotes the margin, we set $\epsilon = 0.03$ for all experiments. Since sentence- and word-level sub-tasks use the same source-target sentences, it is convenient to learn these tasks jointly as follows:

$$L_{\text{QE}} = L_{\text{CE}} + \alpha L_{\text{MSE}} + \beta L_{\text{Rank}}. \tag{5}$$

We use the same loss Eq. 5 for both pre-training and fine-tuning. When pre-training, we use the pseudo data as mentioned above. For fine-tuning, we also explore PE annotation data of the previous years for the MQM sub-task (EN-DE language pair). Target side word-level errors of PE annotation consist of two types of labels: word tags and gap tags (labeled BAD if one or more words should be inserted in between two words). Word tags can be directly converted to MQM tags. To convert gap tags, we label the right word as BAD if the gap tag is BAD. For sentence-level, we normalize the PE sentence scores using the z-score. We mix the PE data and MQM data and use them to fine-tune the QE model.

3.3 Ensemble

We ensemble sentence-level results by averaging all output scores and ensemble word-level results by voting. We also train some models to predict MQM scores without normalization for the EN-DE language pair. To ensemble these models trained with different sentence-level scores, we propose calculating their z-scores and then averaging all z-scores as the ensemble result.

4 Experiments

4.1 Data and Pre-processing

For training the generators and generating pseudo data, we use several parallel data sets. We use the parallel data provided by the WMT translation task [2] for EN-DE(9M), EN-RU(3M), and ZH-EN(3M) language pairs. We use 660K parallel data from OPUS[3] for the KM-EN language pair. Besides, 3.6M parallel data from the target translation model[4] are used for the EN-MR language pair. The PE data used for the EN-DE language pair are provided by WMT2017, WMT2019, and WMT2020.

For pseudo data generation, we learn the BPE vocabulary (Sennrich et al., 2016) with 30K steps using parallel data from each language pair. We can directly use the vocabulary of the XLMR model [5] for pre-training and fine-tuning.

4.2 Implementation and Hyper-parameters

We implement our system with the open source toolkit Fairseq(-py) (Ott et al., 2019). All experiments were conducted on NVIDIA V100 GPUs. Using grid search, we search hyper-parameters (learning rate, weights for different losses). We

[2] https://www.statmt.org/wmt21/translation-task.html

[3] https://opus.nlpl.eu/

[4] https://indicnlp.ai4bharat.org/indic-trans/

[5] https://dl.fbaipublicfiles.com/fairseq/models/xlmr.large.tar.gz

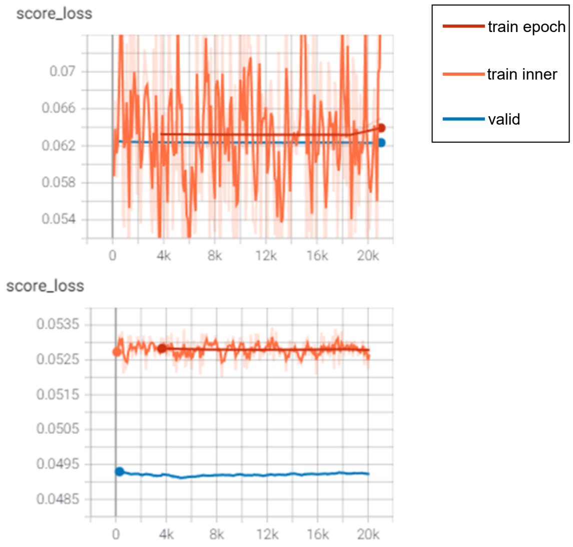


Figure 2: MSE score loss with z-score labels (above); MSE score loss with MQM labels (bottom).

Data	Loss	Spearman
Real	w/o rank	37.88
MLM + Real	w/o rank	43.64
MLM + Real	w/ rank	44.05

Table 3: Results on the validation set of WMT2022 QE EN-DE task. MLM denotes the pseudo data generated by the conditional masked language model.

perform early stopping if the performance does not improve for the last 20 runs.

4.3 Results

We summarize our main results on the test set in Table 2. Our system obtains competitive results over different annotation and language pairs. Especially when we use all techniques proposed in this paper, we finished 1st at both sentence- and word-level on the EN-DE pair.

4.4 Analysis

We conduct preliminary experiments on sentence-level EN-DE sub-task to better reveal the factors that contribute to the performance. Note that we search hyper-parameters with a different scale between different analyses. Thus only results in the same table are comparable.

As shown in Table 3, our pseudo data significantly improve the performance over the baseline. Besides, the rank loss can further improve performance. Table 4 shows that the neural machine translation model is better than the conditional masked language model for generating the pseudo data. Moreover, PE data from WMT2019 is helpful for the MQM task. Surprisingly, PE data from WMT2020 and WMT2017 do not further improve the results. That may be because there are more errors in translations from WMT2020, and the translations from WMT2017 are generated by a statistical machine translation system. We also find that models trained with the MQM scores are better than these using z-scores, shown in Table 5. The MSE score loss seems more stable when using the MQM label, as shown in Figure 2.

Data	Spearman
MLM + Real	49.21
NMT + Real	51.01
MLM + WMT19 + Real	50.45
NMT + WMT19 + Real	51.37
NMT + WMT19,20 + Real	51.15
NMT + WMT19,20,17 + Real	51.24

Table 4: Results on the validation set of WMT2022 QE EN-DE task. NMT denotes the pseudo data generated by the neural machine translation model. WMT## denotes the PE data from WMT20##.

Data	Label	Spearman
NMT + Real	z-score	51.01
NMT + Real	MQM	52.80

Table 5: Results on the validation set of WMT2022 QE EN-DE task with different labels.

5 Conclusion

We present NJUNLP's work to the WMT 2022 Shared Task on Quality Estimation. We propose several novel pseudo data generation methods to bridge the gaps between existing pseudo data and real QE data. To learn the ordinal information, we extend multi-task learning for the QE task with the rank task. We also explore the PE data for the MQM annotation sub-task and propose to ensemble output scores with different scales using the z-score. Experiments show that our pseudo data significantly improve the performance over the baseline. Meanwhile, rank loss and PE data do help. In future research, we will conduct more ablation studies to reveal the contributions of each part.

References

Aljoscha Burchardt and Arle Lommel. 2014. Practical guidelines for the use of mqm in scientific research on translation quality. *Preparation and Launch of a Large-scale Action for Quality Translation Technology, report*, page 19.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Qu Cui, Shujian Huang, Jiahuan Li, Xiang Geng, Zaixiang Zheng, Guoping Huang, and Jiajun Chen. 2021. Directqe: Direct pretraining for machine translation quality estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12719–12727.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kai Fan, Jiayi Wang, Bo Li, Fengming Zhou, Boxing Chen, and Luo Si. 2019. "bilingual expert" can find translation errors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6367–6374.

Marina Fomicheva, Shuo Sun, Erick Fonseca, Chrysoula Zerva, Frédéric Blain, Vishrav Chaudhary, Francisco Guzmán, Nina Lopatina, Lucia Specia, and André FT Martins. 2020. Mlqe-pe: A multilingual quality estimation and post-editing dataset. *arXiv preprint arXiv:2010.04480*.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. TransQuest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5070–5081, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200. Cambridge, MA.

Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold. 2018. Quality estimation for machine translation. *Synthesis Lectures on Human Language Technologies*, 11(1):1–162.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems 30*.

Chrysoula Zerva, Frédéric Blain, Ricardo Rei, Piyawat Lertvittayakumjorn, José G. C. de Souza, Steffen Eger, Diptesh Kanojia, Duarte Alves, Constantin Orăsan, Marina Fomicheva, André F. T. Martins, and Lucia Specia. 2022. Findings of the wmt 2022 shared task on quality estimation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

BJTU-Toshiba's Submission to WMT22 Quality Estimation Shared Task

Hui Huang† **Hui Di**‡ **Chunyou Li**† **Hanming Wu**† **Kazushige Oushi**‡
Yufeng Chen† **Jian Liu**† **Jin'an Xu**†
†Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University
‡Research&Development Center, Toshiba (China) Co., Ltd.
`{18112023, 21120368, 21120416, chenyf, jianliu, jaxu}@bjtu.edu.cn`
`dihui@toshiba.com.cn, kazushige.ouchi@toshiba.co.jp`

Abstract

This paper presents the BJTU-Toshiba joint submission for WMT 2022 quality estimation shared task. We only participate in Task 1 (quality prediction) of the shared task, focusing on the sentence-level MQM prediction. The techniques we experimented with include the integration of monolingual language models and the pre-finetuning of pre-trained representations. We tried two styles of pre-finetuning, namely Translation Language Modeling and Replaced Token Detection. We demonstrate the competitiveness of our system compared to the widely adopted XLM-RoBERTa baseline. Our system is also the top-ranking system on the Sentence-level MQM Prediction for the English-German language pair[1].

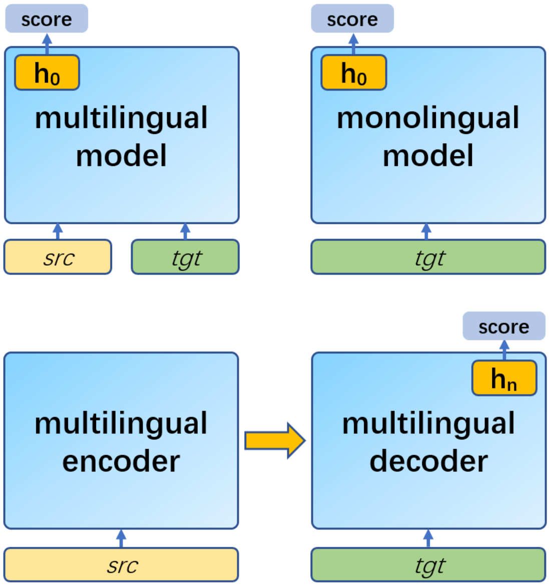


Figure 1: The three QE architectures we adopted.

1 Introduction

Machine translation Quality Estimation (QE) aims to evaluate the quality of machine translation automatically without reference. Compared with commonly used machine translation metrics such as BLEU (Papineni et al., 2002), QE can be applicable to the case where references are unavailable. It has a wide range of applications in post-editing and quality control for machine translation.

This paper introduces in detail the joint submission of Beijing Jiaotong University and Toshiba (China) Corporation to the quality estimation shared task in the 7th Conference on Machine Translation (WMT22), and we mainly focus on the Task 1: quality prediction. This year, the quality prediction task consists of two annotations (DA and MQM) and two levels (sentence-level and word-level), and we only participate in the Sentence-level MQM prediction, of which the goal is to predict the MQM score (Freitag et al., 2021) for each source-target sentence pair. Three language pairs are involved: English-German, Chinese-English and English-Russian, with roughly 10K-20k training pairs provided for each direction.

Our system is mainly based on the ensemble of multiple pre-trained models, both monolingual and multilingual. The monolingual models receive only the target sequence to perform regression (only estimating the target fluency). The multilingual models receive both the source and target sequence to perform regression. We also use in-domain parallel data to pre-finetune the pre-trained models, to adapt their representations to the target language and domain. We try two styles of pre-finetuning, namely Translation Language Model (TLM) and Replaced Token Detection (RTD). The translation language model is to predict the random masked tokens based on the concatenation of source-target pairs. The RTD is to first randomly replace some tokens by another generator, then to detect which token is replaced. Different models are ensembled to get further improvement.

[1] Our codes are openly available at the public repository https://github.com/HuihuiChyan/AwesomeQE.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 621 - 626
December 7-8, 2022 ©2022 Association for Computational Linguistics

Direction	Model	Type	Input	Spearman	Pearson
En-De	mBERT	multilingual understanding	*src-tgt*	0.3621	0.3484
	XLM	multilingual understanding	*src-tgt*	0.3692	0.3682
	XLMR-large	multilingual understanding	*src-tgt*	0.4548	0.4235
	mBART	multilingual encoder-decoder	*src-tgt*	0.3890	0.3946
	OpusMT	multilingual encoder-decoder	*src-tgt*	0.3981	0.4184
	BERT-base	monolingual understanding	*tgt*	0.4620	0.4381
	BERT-large	monolingual understanding	*tgt*	0.4963	0.4574
	Electra-base	monolingual understanding	*tgt*	0.5069	0.4654
	Electra-large	monolingual understanding	*tgt*	**0.5413**	0.4974
Zh-En	XLM	multilingual understanding	*src-tgt*	0.2503	0.1494
	XLMR-large	multilingual understanding	*src-tgt*	0.2614	0.1083
	mBERT	multilingual understanding	*src-tgt*	0.2661	0.1439
	mBART	multilingual encoder-decoder	*src-tgt*	0.2332	0.1021
	OpusMT	multilingual encoder-decoder	*src-tgt*	0.2353	0.1196
	Electra-base	monolingual understanding	*tgt*	0.2337	0.1412
	BERT-large	monolingual understanding	*tgt*	0.2425	0.1149
	Roberta-large	monolingual understanding	*tgt*	0.2523	0.0969
	Deberta-large	monolingual understanding	*tgt*	0.2514	0.1024
	Deberta-v3-large	monolingual understanding	*tgt*	0.2714	0.1486
	Electra-large	monolingual understanding	*tgt*	**0.2829**	0.1475
En-Ru	mBERT	multilingual understanding	*src-tgt*	0.3897	0.3744
	XLM	multilingual understanding	*src-tgt*	0.4281	0.4143
	XLMR-large	multilingual understanding	*src-tgt*	0.4502	0.4144
	mBART	multilingual encoder-decoder	*src-tgt*	0.4174	0.4137
	OpusMT	multilingual encoder-decoder	*src-tgt*	0.4207	0.3884
	BERT-base	monolingual understanding	*tgt*	0.4686	0.3964
	BERT-large	monolingual understanding	*tgt*	0.4899	0.4280
	Roberta-large	monolingual understanding	*tgt*	**0.5175**	0.4265

Table 1: Experiment results on the DEV set of multilingual and monolingual baselines. Results are presented in an ascending order with respect to the spearman's ranking correlation coefficient.

2 Methods

2.1 Architecture

In this work, we perform massive comparison between the multilingual models and monolingual models on QE. Our backbone network is based on several multilingual understanding models, including Multilingual BERT (Devlin et al., 2018), XLM (Lample and Conneau, 2019), XLM-RoBERTa (Ruder et al., 2019), etc. Meanwhile, we integrate several monolingual models, including BERT, RoBERTa (Liu et al., 2020b), DeBERTa (He et al., 2021b), DeBERTa-v3 (He et al., 2021a), Electra (Clark et al., 2020), etc. We also perform estimation on multilingual encoder-decoder models, including Multilingual BART (Liu et al., 2020a) and OpusMT (Tiedemann and Thottingal, 2020)[2].

For multilingual understanding models, we feed the concatenation of *src* (source sentence) and *tgt* (machine translated sentence) to the model, and take the first output hidden state for regression. For monolingual understanding models, we simply feed the *tgt* to the model, and take the first output hidden state for regression. For encoder-decoder

[2] To be specific, we use the released models from https://huggingface.co/Helsinki-NLP/opus-mt-en-de, https://huggingface.co/Helsinki-NLP/opus-mt-zh-en, and https://huggingface.co/Helsinki-NLP/opus-mt-en-ru for En-De, Zh-En and En-Ru, respectively

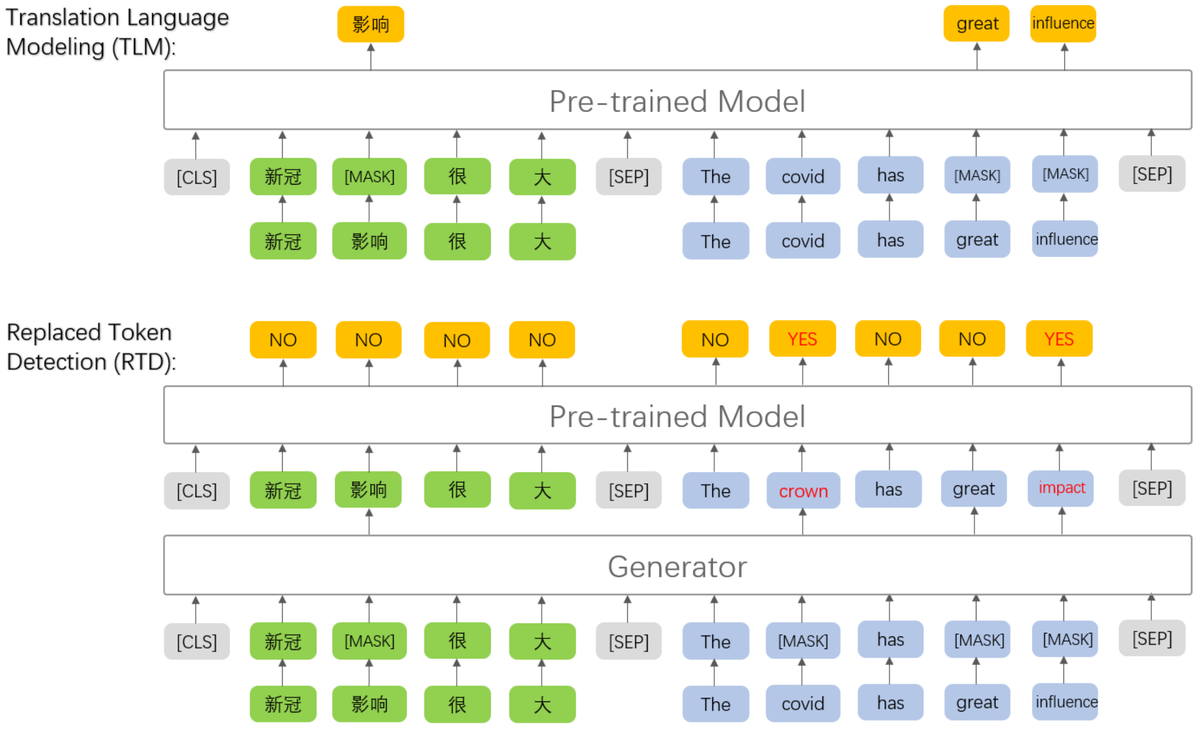


Figure 2: The two different pre-finetuning schemes. Notice for RTD, some masked tokens may be restored correctly by the generator, and we only detect the mismatched tokens.

style models, we feed the *src* to the encoder, the *tgt* to the decoder, and take the last hidden state corresponding to the last token of the *tgt* for regression. All three architectures are depicted in Figure 1.

As shown in Table 1, the monolingual baselines can surpass the multilingual baselines in all directions. Although the alignment information is absent, estimation can still be performed solely on the target text to estimate the fluency. In this year, the MQM prediction data are actually the submissions from the translation evaluation task, therefore most *tgt*s are roughly correct translations aligned with the source sentence, and most translation errors are very subtle. Therefore, it would be easier for the model to estimate the fluency instead of the alignment. With the help of powerful monolingual models, we are able to achieve higher estimation accuracy based solely on the target input.

2.2 Adaptative Pre-finetuning

Fine-tuning pre-trained language models on domain-relevant unlabeled data has become a common strategy to adapt the pretrained parameters to downstream tasks (Gururangan et al., 2020). Previous works also demonstrate the necessity of pre-finetuning when performing QE on pretrained models (Kim et al., 2019; Hu et al., 2020). In this work, we perform two methods to pre-finetune the pre-trained models, namely Translation Language Modeling (TLM) (Lample and Conneau, 2019) and Replaced Token Detection (RTD) (Clark et al., 2020), as shown in Figure 2.

The TLM simply takes the concatenation of parallel sentence pairs as input, and perform masked language modeling. Therefore, when predicting the masked tokens in one side, the model could utilize its context in the parallel side, learning the bilingual alignment.

On the contrary, instead of masking, RTD corrupts the input by replacing some tokens with samples from the output of a smaller masked language model (Specifically, we use the first 1/3 layers of the pre-trained model to initilize the generator). Then the model is trained as a discriminator that predicts for every token whether it is an original or a replacement, learning to distinguish real input tokens from plausible replacements.

Compared with TLM, RTD mainly has three benefits: 1) The corruption procedure solves a mismatch in MLM (or TLM) where the network sees artificial [MASK] tokens during pre-training but not when being fine-tuned on downstream tasks. 2) The loss is calculated on all tokens instead of a subset, therefore improving the pre-finetuning efficiency. 3) The mismatch produced by a language model is more subtle than random masking or replacement, therefore the pre-finetuning naturally fits the final objective, whitch is to detect subtle semantic mismatch.

Direction	Model	Spearman	Pearson
En-De	XLM-R-large	0.4548	0.4235
	w/ TLM	0.5084↑	0.4959
	w/ RTD	0.5109↑	0.5024
	BERT-large	0.4963	0.4574
	w/ TLM	0.5033↑	0.4593
	w/ RTD	0.5127↑	0.4704
	Electra-large	0.5413	0.4974
	w/ TLM	0.4748↓	0.4396
	w/ RTD	0.5220↓	0.4871
	Ensemble	0.5809	0.5313
Zh-En	XLM-R-large	0.2614	0.1083
	w/ TLM	0.2590↓	0.1167
	w/ RTD	0.2888↑	0.1332
	mBERT	0.2661	0.1439
	w/ TLM	0.2912↑	0.1360
	w/ RTD	0.2649↓	0.1254
	Deberta-v3-large	0.2714	0.1486
	w/ TLM	0.2561↓	0.1227
	w/ RTD	0.3076↑	0.1787
	Electra-large	0.2829	0.1475
	w/ TLM	0.2361↓	0.1051
	w/ RTD	0.2493↓	0.1190
	Ensemble	0.3231	0.1692
En-Ru	XLM-R-large	0.4502	0.4144
	w/ TLM	0.4956↑	0.3963
	w/ RTD	0.5092↑	0.3954
	BERT-large	0.4986	0.3964
	w/ TLM	0.5030↑	0.4189
	w/ RTD	0.5170↑	0.4453
	Roberta-large	0.5175	0.4265
	w/ TLM	0.5129↓	0.3979
	w/ RTD	0.5321↑	0.4171
	Ensemble	0.5799	0.4544

Table 2: Experiment results on the DEV set of different pre-finetuning methods and ensemble result.

Direction	Model	Input	Spearman
Zh-En	Deberta-v3-large	*tgt*	0.2892
	Deberta-v3-large	*src-tgt*	0.3076↑
En-Ru	Roberta-large	*tgt*	0.5245
	Roberta-large	*src-tgt*	0.5321↑

Table 3: Experiment results on the DEV set of pre-finetuned models with bilingual or monolingual input.

Both methods are performed on millions of parallel sentence pairs. We firstly train a BERT-based domain classifier to select the in-domain parallel data. Here we use the parallel data from the general translation task of WMT22[3], which contains roughly 20 million pairs for Zh-En and En-De, and 10 million for En-Ru. Specifically, the sentence pairs in the QE training set are deemed as in-domain data, and we randomly sample the same size of data as the general-domain data, and the BERT model is fine-tuned on them as a binary classifier. After that, we select roughly 1 million sentence pairs for each direction.

Notice that for monolingual models we also perform TLM with bilingual input, expecting to introduce further gain with the help of extra information.

As shown in Table 2, both TLM and RTD can improve the estimation accuracy significantly. The multilingual pre-trained model is trained on hundreds of languages simultaneously without any cross-lingual supervision. The monolingual pre-trained model is trained only on the target language. Therefore, adaptation is necessary for both models to solve the language and domain mismatch. Also, the RTD outperforms TLM in most cases, verifying that RTD is more suitable as the pre-finetuning scheme for QE task. Since QE is also targeted at detecting mismatched and disfluent tokens, therefore RTD is more in line with the QE objective.

We also found that after the pre-finetuning step, it would be helpful to feed the bilingual input to the monolingual models, as shown in Table 3. Although monolingual models did not see any text from the source language during pre-training, the knowledge between different languages is transferrable (Artetxe et al., 2020), therefore the fine-tuned model on the target side can also be used to model the semantics of the source side. Besides, subword segmentation also enables the model to represent sequences from unseen language.

The only exception is on Electra, where pre-finetuning brings degradation in all cases. It is possibly because we use the released generator instead

[3] https://www.statmt.org/wmt22/translation-task.html

of using the first few layers to initialize a generator, but it is still confusing why their released generator (which is also used to perform replacement during pre-training stage) would lead to degradation.

2.3 Model Ensemble

Till now, we have obtained different QE models trained with different data and strategies, which can capture different information from the same text. While previous work resort to statistical learning methods to perform model ensemble (Kepler et al., 2019), we think their methods might be overfitting. Therefore, we simple take the average of different predictions (normalized between 0 and 1) as the ensemble result. More specifically, we try different combinations of all available predictions (which are all listed in the Table 2), and make submissions based on the best ensemble result on the DEV set. The performance gain compared to single model is significant as can be seen in Table 2.

3 Conclusion

In this paper, we present our WMT22 QE shared task submission to the sentence-level MQM prediction. We perform massive comparison and demonstrate the effectiveness of monolingual language model. We verify that the pre-trained models can be further improved on target language and target domain via pre-finetuning, and we propose different strategies to pre-finetune the model.

As the machine translation has been developing rapidly, the translation errors current MT system makes have also become more than shallow disalignment. While MT systems are mostly trained with massive parallel data, using the same amount of parallel data to train another QE model seems inefficient, and the monolingual knowledge contained in monolingual models can be more helpful than we expected. While previous work mainly rely on the semantic alignment to perform QE, we think it might be a better option to rely more on monolingual fluency in real applications.

Acknowledgements

The research work descried in this paper has been supported by the National Key RD Program of China (2020AAA0108001) and the National Nature Science Foundation of China (No. 61976015, 61976016, 61876198 and 61370130). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Chi Hu, Hui Liu, Kai Feng, Chen Xu, Nuo Xu, Zefan Zhou, Shiqin Yan, Yingfeng Luo, Chenglong Wang, Xia Meng, Tong Xiao, and Jingbo Zhu. 2020. The niutrans system for the wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1018–1023, Online. Association for Computational Linguistics.

Fabio Kepler, Jonay Trnous, Marcos Treviso, Miguel Vera, Antnio Gis, M. Amin Farajian, Antnio V. Lopes, and Andr F. T. Martins. 2019. Unbabel participation in the wmt19 translation quality estimation shared task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 80–86, Florence, Italy. Association for Computational Linguistics.

Hyun Kim, Joon-Ho Lim, Hyun-Ki Kim, and Seung-Hoon Na. 2019. Qe bert: Bilingual bert using multi-task learning for neural quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 87–91, Florence, Italy. Association for Computational Linguistics.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020a. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Sebastian Ruder, Anders Søgaard, and Ivan Vulić. 2019. Unsupervised cross-lingual representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 31–38, Florence, Italy. Association for Computational Linguistics.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Papago's Submission to the WMT22 Quality Estimation Shared Task

Seunghyun S. Lim[*]
Papago, Naver Corp.
shaun.lim@navercorp.com

Jeonghyeok Park[*]
Papago, Naver Corp.
jeonghyeok.park@navercorp.com

Abstract

This paper describes anonymous submission to the WMT 2022 Quality Estimation shared task. We participate in `Task 1: Quality Prediction` for both sentence and word-level quality prediction tasks. Our system is a multilingual and multi-task model, whereby a single system can infer both sentence and word-level quality on multiple language pairs. Our system's architecture consists of Pretrained Language Model (PLM) and task layers, and is jointly optimized for both sentence and word-level quality prediction tasks using multilingual dataset. We propose novel auxiliary tasks for training and explore diverse sources of additional data to demonstrate further improvements on performance. Through ablation study, we examine the effectiveness of proposed components and find optimal configurations to train our submission systems under each language pair and task settings. Finally, submission systems are trained and inferenced using K-folds ensemble. Our systems greatly outperform task organizer's baseline and achieve comparable performance against other participants' submissions in both sentence and word-level quality prediction tasks.

1 Introduction

Quality Estimation (QE) evaluates the quality of machine translated output without human reference translation (Blatz et al., 2004). Apart from QE models' most obvious usage as being reference-less metrics for MT, it has variety of other applications in Machine Translation (MT) pipeline including but not limited to: parallel corpus filtering (Schwenk et al., 2021), curriculum learning (Ramnath et al., 2021) and decoding (Fernandes et al., 2022).

High performance in both sentence and word-level quality prediction tasks is achieved by incorporating PLM as part of QE model architecture as demonstrated in previous WMT QE findings (Specia et al., 2020, 2021). Previous years' top performers generally incorporate various data augmentation techniques in order to account for limited amount of annotated gold data (Lim et al., 2021; Chen et al., 2021). Multi-task training, ensembling, or incorporating features extracted from external models are few other popular approaches that proved to work well (Lim et al., 2021; Chen et al., 2021; Zerva et al., 2021; Wang et al., 2021a).

Our system is a multilingual and multi-task model, whereby a single system can infer both sentence and word-level quality on multiple language pairs. Our system's architecture (§3.1) consists of PLM and task layers, and is jointly optimized for both sentence and word-level quality prediction tasks (§3.2.1) using multilingual dataset. We propose novel auxiliary tasks (§3.2.2) for training and explore diverse sources of additional data (§3.3) to demonstrate further improvements on performance. Through ablation study (§5), we evaluate each components of our proposed model and use optimal configurations to train our submission systems under each language pair and task settings. Finally, submission systems are trained and inferenced using K-folds ensemble (§3.4.2). Our systems greatly outperform task organizer's baseline and perform very competitively against other participants' submissions in both sentence and word-level quality prediction tasks.

2 Quality Prediction Task and Dataset

In this section we briefly overview two subtasks and their datasets in `Task 1`. Apart from provided `Gold` data as described below, participants are allowed to leverage additional sources of data.

2.1 Sentence Level Quality Prediction

The goal of sentence-level quality prediction is to predict the quality score for each *(source, hypothesis)* sentence pair. Participants are provided with two types of sentence-level quality prediction

[*] These authors contributed equally to this work

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 627 - 633
December 7-8, 2022 ©2022 Association for Computational Linguistics

data depending on how annotations are created: Multi-dimensional Quality Metrics (MQM)[1] and Direct Assessments (DA)[2]. All three language pairs in MQM, and En-Mr in DA are supervised, while remaining four language pairs in DA are unsupervised. Submission systems are evaluated on aforementioned eight language pairs and one surprise language pair[3]. Note that MQM scores are inverted in order to align MQM scores with DA scores.

2.2 Word Level Quality Prediction

The goal of word-level quality prediction is to predict translations errors, assigning OK/BAD tags to each word in hypothesis, given *(source, hypothesis)* sentence pairs. Word-level tags are provided for language pairs same as in sentence-level task, and tags are derived from either MQM annotations (MQM) or post-edited sentences (DA).

3 Approach

Below we describe relevant components of our proposed QE model.

3.1 Model Architecture

Our system employs the Predictor-Estimator architecture (Kim et al., 2017). For our **predictor** we use a PLM, and our choice of PLM is XLM-RoBERTa-large (Conneau et al., 2020) due to its impressive performance on crosslingual downstream tasks. Given source sentence src^X in language X and target sentence tgt^Y in language Y, the concatenation of src^X and tgt^Y are fed as input to the PLM and feature vectors relevant to each task are then passed as inputs to the **estimator**. We utilize four independent 2-layer feed-forward networks as estimators, which are 1024 and 200 dimensions, and are stacked in parallel above PLM. The Predictor-Estimator architecture can be described as:

$$\begin{aligned}
&f(src^x, tgt^y)\\
&= H_{sent}, H_{word}, H_{sentaux}, H_{wordaux}\\
&V_{sent} = \phi_{sent}(H_{sent})\\
&V_{word} = \phi_{word}(H_{word})\\
&V_{sentaux} = \phi_{sentaux}(H_{sentaux})\\
&V_{wordaux} = \phi_{wordaux}(H_{wordaux}),
\end{aligned} \tag{1}$$

where f, H, ϕ, and V are predictor, feature extracted from predictor, estimator, and our final prediction, respectively. We describe H, ϕ, and their corresponding training objectives in §3.2.

3.2 Training Objective

The full training objective of our QE model is shown below in equation (2),

$$\begin{aligned}
\mathcal{L} = &(w_{sent} \cdot \mathcal{L}_{sent}\\
&+ (1 - w_{sent}) \cdot \mathcal{L}_{sentaux}) +\\
&(w_{word} \cdot \mathcal{L}_{word}\\
&+ (1 - w_{word}) \cdot \mathcal{L}_{wordaux}).
\end{aligned} \tag{2}$$

$\mathcal{L}$ and w denote loss functions and loss weight values. w_{sent} and w_{word} are 0.6 and 0.7 respectively. We describe each loss function components in the following subsections.

3.2.1 Multi-task Training

To build a system that is capable of predicting both sentence and word-level quality, our proposed training objective optimizes for $\mathcal{L}_{sent}$ and $\mathcal{L}_{word}$ jointly as shown in equation (2). We use mean squared error (MSE) and weighted cross entropy loss[4] as loss functions for $\mathcal{L}_{sent}$ and $\mathcal{L}_{word}$ respectively. Therefore, ϕ_{sent} is a classification layer with input H_{sent}, which is PLM's last layer [CLS] representation; ϕ_{word} is a classification layer with input H_{word}, which is created by mean pooling PLM's last layer token hidden states[5]. Since sentence and word-level quality prediction tasks are two closely related tasks, we assume some level of transferability of task knowledge between the two when jointly trained.

3.2.2 Auxiliary-task Training

Auxiliary-tasks are additional objectives that are jointly optimized with losses described in §3.2.1.

[1] English-Russian (En-Ru), English-German En-De), and Chinese-English (Zh-En)

[2] English-Marathi (En-Mr), English-Czech (En-Cs), English-Japanese (En-Ja), Khmer-English (Km-En), and Pashto-English (Ps-En)

[3] English-Yoruba (En-Yo), where no train and development data is provided at all

[4] In order to reduce the problem of label imbalance between OK and BAD, we use weighted cross entropy with ratio of OK:BAD = 1:3

[5] If $word_n$ spans $token_{i:j}$, then $H_{word_n} = mean(H_{token_i}, ..., H_{token_j})$

Task	Train Data	Train Method	Train Objective	En-De	En-Ru	Zh-En	En-Mr	Km-En	Ps-En	En-Ja	En-Cs	Multi
Sent	Gold	Vanilla	Single	0.490	0.483	**0.283**	0.551	0.621	0.606	0.315	0.539	0.556
			Multi (3.2.1)	0.493	**0.529**	0.261	0.552	**0.644**	0.614	0.301	0.548	0.562
			Multi+Aux (3.2.1, 3.2.2)	**0.499**	0.516	0.252	**0.555**	0.633	**0.617**	**0.319**	**0.570**	**0.573**
	Gold	Vanilla	Multi + Aux	0.499	0.516	0.252	0.555	0.633	0.617	0.319	0.570	0.573
	Augmented (§3.3)	Vanilla		0.550	0.575	0.274	0.563	0.618	0.611	0.350	0.582	0.609
	Augmented	K-folds ensemble (§3.4.2)		**0.576**	**0.584**	**0.287**	**0.682**	**0.639**	**0.627**	**0.385**	**0.594**	**0.611**
Word	Gold	Vanilla	Single	**0.238**	0.381	**0.208**	0.332	0.404	**0.375**	**0.186**	**0.383**	0.487
			Multi	0.220	0.378	0.201	0.339	0.439	0.367	0.174	0.367	**0.494**
			Multi+Aux	0.229	**0.386**	0.198	**0.358**	**0.455**	0.343	0.169	0.366	0.476
	Gold	Vanilla	Multi + Aux	0.229	**0.386**	0.198	0.358	0.455	0.343	0.169	0.366	0.476
	Augmented	Vanilla		0.285	0.363	0.397	0.443	0.412	0.351	0.153	0.332	0.507
	Augmented	K-folds ensemble (§3.4.2)		**0.301**	0.380	**0.413**	**0.459**	**0.488**	**0.378**	**0.234**	**0.421**	**0.531**

Table 1: Ablation on Train Data, Train Method, and Train Objective. Multi column contains development portion of Gold for all 14 language pairs.

Our intuition is that quality prediction is inherently a complex task even for humans such that human-labels may contain noise. Hence, we appropriately craft original gold labels into secondary labels and use those labels during training as additional learning signals. We expect that training with auxiliary labels can make training more robust and produce a model that is more generalizable.

Sentence-level auxiliary task is a classification task and labels are made as follows: given the n_{th} train set sample's z-standardized score $score_n$, we scale $score_n$ by applying min-max normalization and assign bin (class) labels to each sample. For our experiments, the number of bins is set to 10. Note that min-max scaling is applied to each language pair dataset in order to account for different scales of $score_n$ per dataset. $\phi_{sentaux}$ is a regression layer with input $H_{sentaux}$, which is PLM's last layer [CLS] representation. Likewise, **word-level auxiliary task** is also a classification task and labels are made as follows: given a sample's word-level tags, a sample is assigned to BAD if there exists at least one BAD tags in word-level tags, else OK. $\phi_{wordaux}$ is a classification layer with input $H_{wordaux}$, which is created by mean pooling PLM's last layer token hidden states, excluding special tokens.

3.3 Data Augmentation

We augment training data with additional data, which can be categorized as follows: task-related or pseudo-generated. **Task-related** data are open source data of other downstream tasks, but are similar or can be useful to quality prediction task. We collect data from previous years' WMT Metrics Shared Task[6] and WMT APE Task[7]. Since WMT Metrics Shared Task data contain human DA scores for *(source, hypothesis)* pairs, and WMT APE Task data contain *(source, hypothesis, post-edited)* triplets such that word-level quality annotations can be built using provided word label tagging conventions[8], sentence or word quality labels for this dataset type can be considered high quality.

Pseudo-generated data first assumes bitext[9], *(source, reference)* pairs. We then use NMT models provided by organizers to create *(source, reference, hypothesis)* triplets. Sentence quality labels are generated using COMET[10], which is an open source reference-less QE model. Word quality labels are generated adhering to word label tagging conventions. Labels for pseudo-generated data are considered less accurate compared to task-related data since either labels are pseudo-generated via external model instead being human generated (sentence-level) or do not use actual *(hypothesis, post-edited)* pairs to compute labels (word-level). Refer to Appendix A for detailed list of augmented data.

3.4 Final Model Training

3.4.1 Optimized Configuration

Although we can submit a single model for all language pairs because all our models are multilingual, we submit optimized models for each language pair and task submissions. This is done by choosing and training with optimal configuration for each language pair and task as found in our ablation study (§5, Table 1) or summarized in Appendix B. Our final submissions are optimized for three configurations: train data, train objective, and train method. We further explain each configurations in detail below.

[6] WMT17-21 Metrics Shared Task

[7] WMT16-21 APE Shared Task

[8] https://github.com/deep-spin/qe-corpus-builder#1

[9] Sources of bitext are Europarl, OPUS, Tatoeba and WMT News Translation Task

[10] wmt21-comet-qe-da, https://github.com/Unbabel/COMET

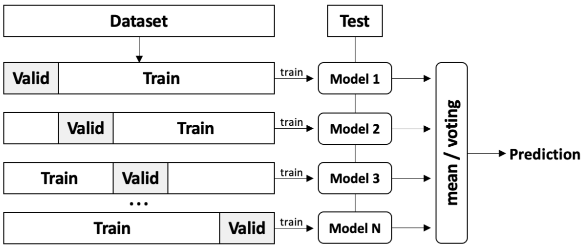


Figure 1: K-folds Ensemble

We have two sources of **train data**: Gold which is data provided by task organizers, and Additional as described in §3.3. This leads us to experiment on two different compositions of train data: Gold and Augmented, where the latter is the aggregation of Gold and Additional.

There are three variants of **train objective**: Single, Multi, and Multi+Aux. Single refers to models that are trained with a single-task objective, either being $\mathcal{L}_{sent}$ or $\mathcal{L}_{word}$. Multi are multi-task models that are trained jointly on both sentence and word-level quality prediction objectives $\mathcal{L}_{sent} + L_{word}$, as described in §3.2.1. Multi+Aux refers to models that are trained with multi-task and auxiliary objectives, as described in equation (2).

For models trained with Vanilla **train method**, we can train with variants of train data but always select best checkpoint using the development set portion of Gold. We explore advanced training methods such as K-folds ensemble (§3.4.2) to further improve model performance.

3.4.2 K-folds Ensemble

As demonstrated in Figure 1, K-folds ensemble (Domingo et al., 2022) distributes the dataset in different training and validation folds such that each individual model uses discrete dataset for training and validation. Compared to vanilla ensembles, where all models are trained using same train data, we expect this method to generate more robust final predictions and become less over-fitted to validation data.

Given a complete set of data[11], we randomly select 1,000 samples for each supervised language pair Gold dataset to create validation set, while the rest are used for training. We repeat this process N=5 times with the constraint that N=5 mutually exclusive validation sets are created. We then train and select best checkpoints for each partition using discrete datasets. An ensemble of N=5 best models, one from each partition, are taken to make final predictions. Prediction mean and majority voting is used for sentence-level and word-level quality prediction respectively.

4 Settings

For all training phases and experiments, we train our model in data parallel on multiple NVIDIA Tesla V100 GPUs for maximum 10 epochs with batch size of 16 and is optimized with Adam (Kingma and Ba, 2015) with a learning rate of $7e^{-6}$. Our implementation is based on PyTorch[12] framework.

All models trained within the scope of this paper are multilingual QE models. We concatenate dataset of all individual language pairs to create a multilingual train dataset for both sentence and word-level quality prediction tasks. We apply the same for development set and always perform model selection using a multilingual dataset[13].

5 Ablation

In this section, we present ablation study of individual components to our model described in §3. All evaluations for ablation in Table 2 are conducted on development portion of Gold.

5.1 Does multilingual training help?

Task	Language	En-De	En-Ru	En-Mr	Ne-En	Si-En
Sent	Single	0.491	0.399	**0.560**	0.798	0.538
	Multi	**0.499**	**0.516**	0.555	**0.805**	**0.550**
Word	Single	0.225	0.306	**0.360**	0.438	0.408
	Multi	**0.229**	**0.386**	0.358	**0.469**	**0.452**

Table 2: Ablation on multilingual training

Training multiple language pairs in a single model through parameter sharing can significantly reduce the cost of model training and maintenance compared with training multiple separate models (Wang et al., 2021b) in Neural Machine Translation. Moreover, we argue that multilingual QE models can collectively learn knowledge from multiple language pairs, which can be particularly be useful in this shared task scenario considering limited training data available per language pair. Table 2 compares the performance of single language pair

[11] We concatenate train and development portion in the case of Gold

[12] https://pytorch.org/

[13] Checkpoints for our final submissions (Table 3, 4) are selected base on performance on multilingual dataset

QE models to multilingual QE models. We see that the performance of multi models are higher than or similar to the performance of single models. Since the performance of multilingual models are at least on par with separate models, this motivates us to use multilingual training when considering additional training and parameter costs of maintaining multiple separate models.

5.2 Does multi-task or auxiliary-task training help?

Row 1 to 3 and 7 to 9 in Table 1 demonstrates ablation on different training objectives. For sentence-level quality prediction tasks, we see that the performance of Multi or Multi+Aux is higher than that of single in all cases except for Zh-En. We observe that adding auxiliary tasks to multi-tasking can give further improvements in most cases. In general, we argue that multi-tasking or adding auxiliary tasks help improve performance of sentence-level QE models. The results for word-level quality prediction tasks are a bit mixed; single achieves highest performance compared to adding any additional training tasks at all for En-De, Zh-En, Ps-En, En-Ja, and En-Cs. We conjecture that multi-tasking or adding auxiliary tasks do not help in word-level as much as they do in sentence-level quality predictions tasks.

5.3 How does train data and train method impact performance?

Row 4 to 6 and 10 to 12 in Table 1 demonstrates ablation on train data and train method. Using additional data (i.e Augmented) improves over Gold in most cases, confirming the importance of using augmented data for quality prediction tasks which mostly are low-resource condition. K-folds ensemble[14] further improves over Vanilla in all cases, again confirming the widely accepted fact that ensembling techniques are useful to give additional boost in performance.

6 Results

Table 3 and 4 demonstrate our final submission systems for sentence-level and word-level quality prediction task respectively. Refer to Appendix B for detailed configurations used for each final submission models.

	Our Submission		Organizer's Baseline	
	Spearman	Rank	Spearman	Rank
Multi	0.4490	4th	0.3172	6th
En-De	0.5815	3rd	0.4548	10th
En-Ru	0.4963	3rd	0.3327	11th
Zh-En	0.3254	4th	0.1641	11th
Multi	0.5015	2nd	0.4148	5th
Multi (w/o En-Yo)	0.5710	3rd	0.4974	6th
En-Mr	0.6038	1st	0.4356	9th
En-Cs	0.6362	2nd	0.5598	7th
En-Ja	0.3266	4th	0.2716	9th
Km-En	0.6526	3rd	0.5788	7th
Ps-En	0.6713	3rd	0.6410	6th
# params	560M		564M	
Disk space	2,243MB		2,280MB	

Table 3: Submission results on Sentence-level Quality Prediction Task

	Our Submission		Organizer's Baseline	
	MCC	Rank	MCC	Rank
Multi	0.3167	2nd	0.2345	3rd
Multi (w/o En-Yo)	0.3431	2nd	0.2569	3rd
En-De	0.3186	2nd	0.1824	5th
En-Ru	0.4207	2nd	0.2027	5th
Zh-En	0.3514	2nd	0.1036	5th
En-Mr	0.4178	1st	0.3058	5th
En-Cs	0.3961	3rd	0.3245	4rd
En-Ja	0.2573	2nd	0.1751	4th
Km-En	0.4291	1st	0.4016	4th
Ps-En	0.3735	2nd	0.3593	3rd
# params	560M		564M	
Disk space	2,243MB		2,280MB	

Table 4: Submission results on Word-level Quality Prediction Task

7 Conclusions

In this work, we describe our system submission to the WMT 2022 Quality Estimation shared task. Our system is a multilingual and multi-task model for both sentence and word level quality prediction tasks. We demonstrate through ablation study that additional training objectives and data can further improve quality prediction performance. Our final model is trained and inferenced using K-folds ensemble which show remarkable performance in all language pairs and tasks. However, we find that multi-task or auxiliary-task training do not help in word-level as much as they do in sentence-level quality prediction. Further analysis to understand the dynamics of training with multiple objectives and improvements on word-level quality prediction are challenges that we need to overcome in future work.

[14] We leave out development portion of Gold for final evaluation within the scope of ablation

References

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 315–321, Geneva, Switzerland. COLING.

Yimeng Chen, Chang Su, Yingtao Zhang, Yuxia Wang, Xiang Geng, Hao Yang, Shimin Tao, Guo Jiaxin, Wang Minghan, Min Zhang, Yujia Liu, and Shujian Huang. 2021. HW-TSC's participation at WMT 2021 quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 890–896, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jaime Duque Domingo, Roberto Medina Aparicio, and Luis Miguel Gonzx00E1;lez Rodrigo. 2022. Cross validation voting for improving cnn classification in grocery products. *IEEE Access*, 10:20913–20925.

Patrick Fernandes, António Farinhas, Ricardo Rei, José G. C. de Souza, Perez Ogayo, Graham Neubig, and André F. T. Martins. 2022. Quality-aware decoding for neural machine translation.

Hyun Kim, Hun-Young Jung, Hongseok Kwon, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator: Neural quality estimation based on target word prediction for machine translation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 17(1).

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Seunghyun Lim, Hantae Kim, and Hyunjoong Kim. 2021. Papago's submission for the WMT21 quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 935–940, Online. Association for Computational Linguistics.

Sahana Ramnath, Melvin Johnson, Abhirut Gupta, and Aravindan Raghuveer. 2021. HintedBT: Augmenting Back-Translation with quality and transliteration hints. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1717–1733, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500, Online. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the WMT 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the WMT 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 684–725, Online. Association for Computational Linguistics.

Jiayi Wang, Ke Wang, Boxing Chen, Yu Zhao, Weihua Luo, and Yuqi Zhang. 2021a. Qemind: Alibaba's submission to the WMT21 quality estimation shared task. *CoRR*, abs/2112.14890.

Rui Wang, Xu Tan, Renqian Luo, Tao Qin, and Tie-Yan Liu. 2021b. A survey on low-resource neural machine translation.

Chrysoula Zerva, Daan van Stigt, Ricardo Rei, Ana C Farinha, Pedro Ramos, José G. C. de Souza, Taisiya Glushkova, Miguel Vera, Fabio Kepler, and André F. T. Martins. 2021. IST-unbabel 2021 submission for the quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 961–972, Online. Association for Computational Linguistics.

A Data Augmentation

Augment Data Type	Label Type	Language	# of samples	Original Source
Pseudo-generated	Sent, Word	En-De	500,000	Europarl, OPUS, Tatoeba, and WMT News Translation Task
		En-Ru	500,000	
		Zh-En	500,000	
		En-Mr	500,000	
		Km-En	423,583	
		Ps-En	131,163	
		En-Ja	500,000	
		En-Cs	500,000	
Task-related	Sent	En-De	78,616	WMT Metrics Shared Task Data
		En-Ru	72,024	
		Zh-En	136,938	
		Km-En	4,722	
		Ps-En	4,611	
		En-Ja	24,429	
		En-Cs	70,911	
		En-Zh	94,667	
		De-En	109,907	
		Ru-En	70,276	
		Ja-En	23,399	
	Word	En-De	37,000	WMT APE Shared Task Data
		En-Ru	17,112	
		En-Mr	19,000	
		De-En *	28,000	
		En-Zh *	9,000	

Table 5: Details on augmented data

B Optimal Configuration

Task	Label Type	Language Pair	Train Objective	Train Data
Sent	MQM	Multi	Multi+Aux	Gold, Task-related
		En-De	Multi+Aux	Gold, Task-related
		En-Ru	Multi	Gold, Task-related
		Zh-En	Single	Gold, Task-related
	DA	Multi	Multi+Aux	Gold, Task-related
		Multi (w/o En-Yo)	Multi+Aux	Gold, Task-related
		En-Mr	Multi+Aux	Gold, Task-related
		En-Cs	Multi	Gold, Task-related
		En-Ja	Multi+Aux	Gold, Task-related
		Km-En	Multi+Aux	Gold, Task-related
		Ps-En	Multi+Aux	Gold, Task-related
Word	MQM + DA	Multi	Multi	Gold, Task-related, Pseudo-generated
		Multi (w/o En-Yo)	Multi	Gold, Task-related, Pseudo-generated
	MQM	En-De	Single	Gold, Task-related
		En-Ru	Multi+Aux	Gold, Task-related
		Zh-En	Single	Gold, Task-related
	DA	En-Mr	Multi+Aux	Gold, Task-related
		En-Cs	Single	Gold, Task-related, Pseudo-generated
		En-Ja	Single	Gold, Task-related, Pseudo-generated
		Km-En	Multi+Aux	Gold, Task-related, Pseudo-generated
		Ps-En	Single	Gold, Task-related, Pseudo-generated

Table 6: Details on optimal configuration

COMETKIWI: IST-Unbabel 2022 Submission for the Quality Estimation Shared Task

Ricardo Rei[*1,2,4], **Marcos Treviso**[*3,4], **Nuno M. Guerreiro**[*3,4], **Chrysoula Zerva**[*3,4],
Ana C. Farinha[1], **Christine Maroti**[1], **José G. C. de Souza**[1], **Taisiya Glushkova**[3,4],
Duarte M. Alves[1,4], **Alon Lavie**[1], **Luisa Coheur**[2,4], **André F. T. Martins**[1,3,4]
[1]Unbabel, Lisbon, Portugal, [2]INESC-ID, Lisbon, Portugal
[3]Instituto de Telecomunicações, Lisbon, Portugal
[4]Instituto Superior Técnico, University of Lisbon, Portugal

Abstract

We present the joint contribution of IST and Unbabel to the WMT 2022 Shared Task on Quality Estimation (QE). Our team participated on all three subtasks: (i) Sentence and Word-level Quality Prediction; (ii) Explainable QE; and (iii) Critical Error Detection. For all tasks we build on top of the COMET framework, connecting it with the predictor-estimator architecture of OPENKIWI, and equipping it with a word-level sequence tagger and an explanation extractor. Our results suggest that incorporating references during pretraining improves performance across several language pairs on downstream tasks, and that jointly training with sentence and word-level objectives yields a further boost. Furthermore, combining attention and gradient information proved to be the top strategy for extracting good explanations of sentence-level QE models. Overall, our submissions achieved the best results for all three tasks for almost all language pairs by a considerable margin.[1]

1 Introduction

Quality Estimation (QE) is the task of automatically assigning a quality score to a machine translation output without depending on reference translations (Specia et al., 2018). In this paper, we describe the joint contribution of Instituto Superior Técnico (IST) and Unbabel to the WMT22 Quality Estimation shared task (Zerva et al., 2022), where systems were submitted to three tasks: (i) Sentence and Word-level Quality Prediction; (ii) Explainable QE; and (iii) Critical Error Detection.

This year, we leverage the similarity between the tasks of MT evaluation and QE and bring together the strengths of two frameworks, COMET (Rei et al., 2020), which has been originally developed for reference-based MT evaluation, and OPENKIWI (Kepler et al., 2019), which has been developed for word-level and sentence-level QE. Namely, we implement some of the features of the latter, as well as other new features, into the COMET framework. The result is COMETKIWI, which links the predictor-estimator architecture with COMET training-style, and incorporates word-level sequence tagging.

Given that some language pairs (LPs) in the test set were not present in the training data, we aimed at developing QE systems that achieve good multilingual generalization and that are flexible enough to account for unseen languages through few-shot training. To do so, we start by pretraining our QE models on Direct Assessments (DAs) annotations from the previous year's Metrics shared task as it was shown to be beneficial in our previous submission (Zerva et al., 2021). Then we fine-tune our models with the data made available by the shared task.[2] We experimented with different pretrained multilingual transformers as the backbones of our models, and we developed new explainability methods to interpret them. We describe our systems and their training strategies in Section 3. Overall, our main contributions are:

- We combine the strengths of COMET and OPENKIWI, leading to COMETKIWI, a model that adopts COMET training features useful for multilingual generalization along with the predictor-estimator architecture of OPENKIWI.

- Following our previous work (Zerva et al., 2021), we show the importance of pretraining QE models on annotations from the Metrics shared task.

- We show that we can improve results for new LPs with only 500 examples without harming correlations for other LPs.

- We propose a new interpretability method that uses attention and gradient information along

[*] Equal contribution. ✉ ricardo.rei@unbabel.com

[1] https://github.com/Unbabel/COMET

[2] For zero-shot LPs we use 500 training examples which means we turn it into a few-shot setting. The only exception is English→Yoruba which was kept zero-shot.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 634 - 645
December 7-8, 2022 ©2022 Association for Computational Linguistics

with a head-level scalar mix module that further refines the relevance of attention heads.

Our submitted systems achieve the best multilingual results on all tracks by a considerable margin: for sentence-level DA our system achieved a 0.572 Spearman correlation (+7% than the second best system); for word-level our system achieved a 0.341 MCC score (+2.4% than the second best system); and for Explainable QE our system achieved 0.486 R@K score (+10% than the second best system). The official results for all LPs are presented in Table 6 in the appendix.

2 Background

Quality Estimation. QE systems are usually designed according to the granularity in which predictions are made, such as sentence and word-level. In sentence-level QE, the goal is to predict a single quality score $\hat{y} \in \mathbb{R}$ given the whole source and its translation as input. Word-level QE works in a lower granularity level, with the goal of predicting binary quality labels $\hat{y}_i \in \{\text{OK}, \text{BAD}\}$ for all $1 \leq i \leq n$ *machine-translated words*, indicating whether that word is a translation error or not.

Transformers. The multi-head attention mechanism is the key component in transformers, being responsible for contextualizing the information within and across input sentences (Vaswani et al., 2017). Concretely, given as input a matrix $\boldsymbol{Q} \in \mathbb{R}^{n \times d}$ containing d-dimensional representations for n queries, and matrices $\boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{m \times d}$ for m keys and values, the *scaled dot-product attention* at a single head is computed as:

$$\mathsf{att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \pi \underbrace{\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d}} \right)}_{\boldsymbol{Z} \in \mathbb{R}^{n \times m}} \boldsymbol{V} \in \mathbb{R}^{n \times d}. \quad (1)$$

The π transformation maps rows to distributions, with softmax being the most common choice, $\pi(\boldsymbol{Z})_{ij} = \mathsf{softmax}(\boldsymbol{z}_i)_j$. Multi-head attention is computed by evoking Eq. 1 in parallel for each head h:

$$\mathsf{head}_h(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathsf{att}(\boldsymbol{Q}\boldsymbol{W}_h^Q, \boldsymbol{K}\boldsymbol{W}_h^K, \boldsymbol{V}\boldsymbol{W}_h^V),$$

where $\boldsymbol{W}_h^Q$, $\boldsymbol{W}_h^K$, $\boldsymbol{W}_h^V$ are learnable linear transformations. Finally, the output of the multi-head attention module at the ℓ-th layer is a set of hidden states $\boldsymbol{H}_\ell \in \mathbb{R}^{n \times d}$ formed via the concatenation of all $\boldsymbol{h}_{\ell,1}, ..., \boldsymbol{h}_{\ell,H}$ heads in that layer followed by a learnable linear transformation $\boldsymbol{W}^O$:

$$\boldsymbol{H}_\ell = \mathsf{concat}(\boldsymbol{h}_{\ell,1}, ..., \boldsymbol{h}_{\ell,H})\boldsymbol{W}^O.$$

The hidden states are further refined through position-wise feed-forward blocks and residual connections to obtain a final representation: $\boldsymbol{H}_\ell = \mathsf{FFN}(\boldsymbol{H}_\ell) + \boldsymbol{H}_\ell$. Transformers with only encoder-blocks, such as BERT (Devlin et al., 2019) and XLM (Conneau et al., 2020), have only the encoder self-attention, and thus $m = n$.

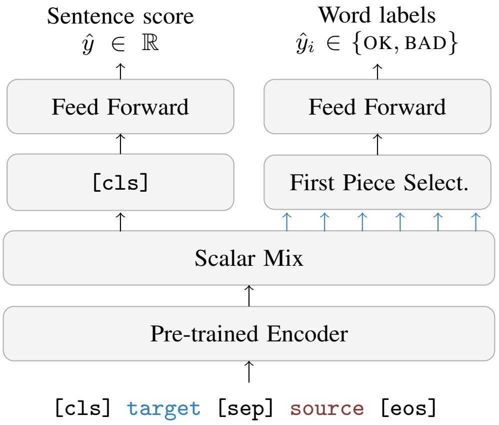


Figure 1: General architecture of COMETKIWI for sentence-level (left part) and word-level QE (right part).

3 Implemented Systems

The overall architecture of our models is shown in Figure 1. The machine translated sentence $\boldsymbol{t} = \langle t_1, ..., t_n \rangle$ and its source sentence counterpart $\boldsymbol{s} = \langle s_1, ..., s_m \rangle$ are concatenated and passed as input to the encoder, which produces d-dimensional hidden state vectors $\boldsymbol{H}_0, ..., \boldsymbol{H}_L$ for each layer $0 \leq \ell \leq L$, where $\boldsymbol{H}_i \in \mathbb{R}^{(n+m) \times d}$, where $\ell = 0$ corresponds to the embedding layer. Next, all hidden states are fed to a scalar mix module (Peters et al., 2018) that learns a weighted sum of the hidden states of each layer of the encoder, producing a new sequence of aggregated hidden states $\boldsymbol{H}_{\text{mix}}$ as follows:

$$\boldsymbol{H}_{\text{mix}} = \lambda \sum_{\ell=0}^{L} \beta_\ell \boldsymbol{H}_\ell, \quad (2)$$

where λ is a scalar trainable parameter, $\boldsymbol{\beta} \in \triangle^L$, is given by $\boldsymbol{\beta} = \mathsf{sparsemax}(\boldsymbol{\phi})$ using a sparse transformation (Martins and Astudillo, 2016), with $\boldsymbol{\phi} \in \mathbb{R}^L$ as learnable parameters and $\triangle^L := \{\boldsymbol{\beta} \in \mathbb{R}^L : \boldsymbol{1}^\top \boldsymbol{\beta} = 1, \boldsymbol{\beta} \geq 0\}$[3].

[3] As it has been shown in (Rei et al., 2022) not all layers are relevant and thus, using sparsemax we learn to ignore layers

For sentence-level models, the hidden state of the first token (`<cls>`) is used as sentence representation $\boldsymbol{H}_{\text{mix},0} \in \mathbb{R}^d$, which, in turn, is passed to a 2-layered feed-forward module in order to get a sentence score prediction $\hat{y} \in \mathbb{R}$. For word-level models, we first retrieve the hidden state vectors associated with the first word piece of each machine translated token, and then pass them to a linear projection to get word-level predictions $\hat{y}_i \in \{\text{OK}, \text{BAD}\}, \forall_{1 \leq i \leq n}$. Moreover, attention matrices $\boldsymbol{A}_{1,1}, ..., \boldsymbol{A}_{L,H}$ for all layers and heads are also recovered as a by-product of the forward propagation.

Pretraining on Metrics Data. Every year, the WMT News Translation shared task organizers collect human judgments in the form of DAs. The collective corpora of 2017, 2018, and 2019 contain 24 LPs and a total of 657k samples with source, target, reference, and DA score. We follow our experiments from last year (Zerva et al., 2021) and start by pretraining our QE models on this data using the learning objective proposed by UniTE (Wan et al., 2022), which incorporates reference translations into training and thus acts as data augmentation.

Setting pretrained transformers as encoders. We follow the recent trend (Kepler et al., 2019; Ranasinghe et al., 2020) and experiment with three different pretrained multilingual transformers as the encoder layer of our models: XLM-R Large (Conneau et al., 2020),[4] InfoXLM Large (Chi et al., 2021),[5] and RemBERT (Chung et al., 2021).[6] XLM-R and InfoXLM consist of 24 encoder blocks with 16 attention heads each, whereas RemBERT has 32 encoder blocks with 18 attention heads each.

3.1 Task 1: Quality prediction

After the pretraining phase, we adapt our models to the released QE data using source and translation (i.e., in this phase we do not include references) to the different type of quality assessments provided, namely, DA and HTER[7] from the MLQE-PE corpus (Fomicheva et al., 2022) and Multidimensional Quality Metrics (MQM) annotations from WMT 2020 and 2021 (Freitag et al., 2021a,b).

3.1.1 Sentence-level quality prediction

For the sentence-level QE task we consider a multi-task setting (using sentence scores alongside supervision from OK/BAD tags) and the sentence-level only setting, with supervision only from the sentence-level quality assessment y. We found that adding the word-level supervision was beneficial for models built on top of InfoXLM. For the sentence-level supervision we used both DA and MQM scores. In this multi-task setting we use a combined loss as described in Eq. 5:

$$\mathcal{L}_{\text{sent}}(\theta) = \frac{1}{2}(y - \hat{y}(\theta))^2 \tag{3}$$

$$\mathcal{L}_{\text{word}}(\theta) = -\frac{1}{n}\sum_{i=1}^{n} w_{y_i} \log p_\theta(y_i) \tag{4}$$

$$\mathcal{L}(\theta) = \lambda_s \mathcal{L}_{\text{sent}}(\theta) + \lambda_w \mathcal{L}_{\text{word}}(\theta), \tag{5}$$

where $w \in \mathbb{R}^2$ represents the class weights given for OK and BAD tags, and λ_s, λ_w are used to weigh the combination of the sentence and word-level losses, respectively. Note that $\lambda_s = 1$ and $\lambda_w = 0$ yields a fully sentence-level model.

Few-shot language adaptation. Since in this shared task submissions are tested on 5 LPs for which there is no official training data (*km-en*, *ps-en*, *en-ja*, *en-cs*, *en-yo*), we experimented with few-shot adaptation using half of the data released in the official development set. The official development set has 1K examples for each language pair (except *en-yo* for which there is no available data). To perform few-shot language adaptation we split the data into two halves: one for fine-tuning and another for validation.

Ensembling models. For our final submission for Direct Assessments we combine six multilingual systems using different hyperparameters by computing an weighted average of their outputs, where the weights for each language pair were tuned with Optuna (Akiba et al., 2019). The major difference between the ensembled models comes from the underlying encoder and whether or not they used word-level supervision. Three models of our final ensemble use word-level supervision while the other three use only sentence-level supervision. Regarding the encoder, three models use InfoXLM, two models use RemBERT and a single model uses XLM-R.

Our final submission for MQM predictions was an ensemble of eleven multilingual systems, which

that do not help in the task at hands

[4] https://huggingface.co/xlm-roberta-large

[5] https://huggingface.co/microsoft/infoxlm-large

[6] https://huggingface.co/google/rembert

[7] HTERs are available only for word-level subtasks.

combined the six systems used in the DA ensemble as well as five additional systems. For these additional systems, we made two major adjustments to the fine-tuning process. First, we filtered the DA data to the languages that were included in the MQM LPs, namely *ru-en*, *en-zh*, and *en-de*. Second, we incorporated the MQM data into the fine-tuning process, either as an additional fine-tuning step after fine-tuning on the language-filtered DA data, or by concatenating the DA and MQM data together. All additional systems used word-level supervision in addition to sentence-level and used InfoXLM as encoder.

3.1.2 Word-level quality prediction

Similarly, for the word-level QE tasks we experimented with both the multi-task setting and word-labels only ($\lambda_s = 0$ and $\lambda_w = 1$). Overall, we found that adding the sentence-level supervision was beneficial, especially for the languages pairs included in the test-set. Nonetheless, for some LPs, ignoring sentence-level supervision showed superior performance. Due to the mix of high-, mid- and low-resource languages in the data, the distribution of OK and BAD tags differs substantially between LPs leading to inconsistent performance in terms of MCC (see Table 5 in the appendix). To mitigate this, for the word-level subtask, we prepend a language prefix token to the beginning of the source and target segments during training and testing.

Pretraining on post-edit corpora. Extending the pretraining on Metrics data, we pretrain the word-level models on two corpora that include both word-level labels and sentence (HTER) scores, namely QT21 (Specia et al., 2017) and APEQuest (Ive et al., 2020). We compute the sentence-level score, using translation edit rate (TER) (Snover et al., 2006) between the target and the corresponding post-edited sentence.

Ensembling models. For word-level we followed a similar ensembling technique used for sentence-level, namely we combine multiple systems trained with different hyperparameters, encoders and pre-training setups. In the case of word-level predictions however, we need to resolve how to aggregate multiple predictions into OK/BAD tags. We use Optuna (Akiba et al., 2019) to choose how to weight and combine the models based on performance for each language pair on our internal test-set and we compare three different approaches:

1. A naive "best-only" approach: we identify the best model for each LP and use its predictions.

2. We ensemble the logits of each model: for each input segment we compute an ensembles of logits as $\sum_{i \in \mathcal{M}} w_i v_i$, where $\mathcal{M}$ is the set of models, w_i is the weight of each model and v_i the model logit vector. We use Optuna to find the optimal weight w_i for each model in each LP.

3. We ensemble the predicted tags of each model: for each input segment we compute an ensembles of tags as $\alpha \sum_{i \in \mathcal{M}} w_i c_i$, where c_i is the predicted class and α is the weight given for the BAD class. We use Optuna to find the optimal weights w_i for each model and the optimal BAD weight α for each LP.

In the final submission we combine five models for the post-edit originated LPs: a RemBERT based model, an InfoXLM based model pretrained on APEQuest and QT21, and three checkpoints that are based on InfoXLM but use different parameters for the BAD/OK weights and learning rate that were found via Optuna. For MQM we also combine five models, but this time instead of choosing three checkpoints based on optimising weights and learning rate, we use three different checkpoints with different training data mix on the relevant DA LPs, as this seemed to impact the performance on MQM word-level more than the weight ratios. Refer to §4 and Table 3 for more details.

3.2 Task 2: Explainable QE

The goal of the Explainable QE task is to identify machine translation errors without relying on word-level label information. In other words, it can be cast as an unsupervised word-level quality estimation problem, where explanations can be seen as highlights, representing the relevance of input words w.r.t. the model's prediction via continuous scores, aiming at identifying tokens that were not properly translated.

Several explainability methods can be used to extract highlights from a sentence-level model, such as post-hoc (Ribeiro et al., 2016; Arras et al., 2016) or inherently interpretable methods (Lei et al., 2016; Guerreiro and Martins, 2021). In our submission, we opted to use attention-based methods as they achieved the best results in the previous constrained track of the Explainable QE shared task (Fomicheva et al., 2021). Concretely, we take inspiration in the method developed by Treviso et al.

	Direct Assessment												
Encoder	**km-en**	**ps-en**	**en-ja**	**en-cs**	**en-mr**	**ru-en**	**ro-en**	**en-zh**	**en-de**	**et-en**	**si-en**	**ne-en**	**avg.**
Baseline (Zerva et al., 2021)													
XLM-R	0.615	0.601	0.295	0.535	0.419	0.703	0.828	0.513	0.500	0.806	0.565	0.793	0.598
Pretrained models													
InfoXLM	0.619	0.603	0.328	0.510	0.462	0.731	0.829	0.554	0.516	0.803	0.561	0.777	0.608
RemBERT	0.600	0.621	0.338	0.525	0.447	0.680	0.818	0.487	0.491	0.810	0.525	0.747	0.591
XLM-R	0.610	0.579	0.325	0.503	0.405	0.715	0.832	0.541	0.514	0.782	0.540	0.740	0.591
Sentence-level only													
XLM-R	0.628	0.591	0.350	0.531	0.551	0.761	0.859	0.577	0.568	0.800	0.565	0.796	0.631
InfoXLM	0.629	0.623	0.348	0.515	0.574	0.747	0.858	0.586	0.551	0.828	0.568	0.790	0.635
RemBERT	0.634	0.631	0.346	0.570	0.564	0.754	0.862	0.534	0.531	0.822	0.550	0.782	0.632
Few-shot Language Adaptation													
XLM-R	0.650	0.619	0.352	0.551	0.546	0.753	0.852	0.571	0.554	0.813	0.562	0.798	0.635
InfoXLM	0.641	0.650	0.367	0.549	0.549	0.751	0.855	0.591	0.565	0.824	0.563	0.803	0.642
RemBERT	0.625	0.641	0.367	0.568	0.563	0.756	0.857	0.540	0.527	0.824	0.568	0.796	0.636
Sentence + word-level training													
InfoXLM	0.617	0.586	0.344	0.532	0.572	0.761	0.865	0.586	0.579	0.829	0.576	0.804	0.637
RemBERT	0.634	0.628	0.356	0.564	0.571	0.762	0.860	0.541	0.553	0.826	0.564	0.799	0.638
Few-shot Language Adaptation													
InfoXLM	0.643	0.632	0.335	0.557	0.560	0.766	0.860	0.575	0.582	0.833	0.578	0.809	0.644
RemBERT	0.644	0.645	0.356	0.567	0.568	0.759	0.856	0.545	0.552	0.835	0.561	0.804	0.641
Final Ensemble													
Ensemble 6x	**0.664**	**0.669**	**0.380**	**0.591**	**0.593**	**0.782**	**0.871**	**0.597**	**0.593**	**0.845**	**0.588**	**0.820**	**0.666**

Table 1: Results for sentence-level QE in terms of Spearman correlation for DA.

(2021), which consists of scaling attention weights by the ℓ_2-norm of value vectors (Kobayashi et al., 2020) and finding the attention heads with the best performance on the dev set, and propose two new modifications:

- **Attention × GradNorm:** Following the findings of Chrysostomou and Aletras (2022), we decided to extract explanations that consider both attention and gradient information. More precisely, we scale the attention weights by the ℓ_2-norm of the gradient of value vectors:

$$\boldsymbol{A}_{\ell,h} \left\| \nabla \boldsymbol{V}_{\ell,h} \right\|_2 . \tag{6}$$

- **Head Mix:** We reformulate the scalar mix module (Eq. 2) to consider different weights for representations coming from different attention heads as follows:

$$\boldsymbol{H}_{\text{mix}} = \lambda \sum_{\ell=0}^{L} \beta_\ell \sum_{h=1}^{H} \gamma_{\ell,h} \boldsymbol{h}_{\ell,h}, \tag{7}$$

where the *layer* mix coefficients $\boldsymbol{\beta} \in \triangle^L$ are given by $\boldsymbol{\beta} = \pi(\boldsymbol{\phi})$, and the *head* mix coefficients $\boldsymbol{\gamma}_\ell \in \triangle^H$ are given by $\boldsymbol{\gamma}_\ell = \pi(\boldsymbol{\theta}_\ell)$. $\lambda \in \mathbb{R}$, $\boldsymbol{\phi} \in \mathbb{R}^L$ and $\boldsymbol{\theta} \in \mathbb{R}^{L \times H}$ are learnable parameters. We experimented both with dense (π as softmax) and sparse (π as sparsemax, Martins and Astudillo 2016) transformations. After training, the Head Mix coefficients can help to find attention heads with high validation performance, which is helpful for explaining zero-shot LPs.

Furthermore, since all of our sentence-level models use subword tokenization, to get explanations for an entire word we follow Treviso et al. (2021) and sum the scores of its word pieces.

Ensembling explanations. In our final submissions we average the explanation scores of different attention heads and layers to create a final explainer. We decided which heads and layers to aggregate together by looking at their performance on the dev set, selecting the top-5 with the highest explainability score.

3.3 Task 3: Critical Error Detection

Critical translations are defined as translations with strongly semantic deviations from the original source sentence, with the potential to lead to negative impacts in critical applications. The goal of this task is to predict sentence-level scores indicating whether a translation contains a critical error. Since the evaluation metrics automatically account for different binarization thresholds to separate good translations from bad ones, for this task we employed a single sentence-level InfoXLM model from Task 1 that was trained on DA data. Moreover, we participated only in the *constrained setting*, meaning that we did not trained our systems specifically for this task. Therefore, our goal for this task was to validate whether our QE system from Task 1 was able to detect and differentiate translations with critical errors.

4 Experimental Results

As we have seen in Section 3, for our experiments we split the provided development sets into two equal size halves creating a new internal devset and an internal testset. The resulting sets contain ≈ 500 segments per language pair for both DA and MQM, word and sentence-level. As for baselines we used our submitted systems from previous shared tasks: for Task 1 we used the M1M-ADAPT (Zerva et al., 2021), and for Task 2 we used the Attn × Norm explainer (Treviso et al., 2021). The official results for Task 1 and Task 2 are shown in Table 6.

4.1 Quality Estimation

Sentence-level submissions were evaluated using the Spearman's rank correlation. Pearson's correlation, MAE, and RMSE were also used as secondary metrics, but here we report only Spearman correlation since it was the primary metric used to rank systems. Word-level submission were evaluated using MCC, F_1-OK, and F_1-BAD, but we report only MCC as it was considered the main metric. The submitted systems were independently evaluated on in-domain and zero-shot LPs for direct assessments and MQM.

Direct Assessments. Results for sentence-level DAs can be seen in Table 1. The results show that the training strategies employed in COMETKIWI, namely (i) pretraining models using Metrics data and (ii) incorporating references into training, lead to a correlation close to our best system from last year while disregarding the data from the MLQE-PE corpus. When fine-tuning on MLQE-PE data, we get overall improvements of $\sim$ 4%, and further fine-tuning on new LPs gives $\sim$ 1% overall improvement. Still, for the unseen LPs (*km-en, ps-en, en-ja, en-cs*), we got improvements between 2-3% with just 500 samples. Among the three backbone transformers, we noticed that InfoXLM is the one that leads to a higher Spearman correlation (+1.7% than XLM-R and RemBERT). Furthermore, including word-level supervision always maintains or improves the results, especially for InfoXLM. In contrast, RemBERT does not seem to benefit from this signal. We suspect that, for this task, the benefit of word-level supervision is not higher because the word-level information is coming from post-editions, which are conceptually different from DA annotations.

	MQM			
System (fine-tuned on)	**en-de**	**en-ru**	**zh-en**	**avg.**
Sentence-level only				
DA	0.529	0.534	0.215	0.426
DA + MQM	0.531	0.552	0.250	0.444
DA (3 LPs) + MQM	0.538	0.550	0.262	0.450
Sentence + word-level training				
DA	0.525	0.557	0.217	0.433
DA (3 LPs)	0.560	0.561	0.222	0.448
DA + MQM	0.540	0.568	0.262	0.457
DA (3 LPs) + MQM	0.553	**0.569**	0.268	0.463
DA (3 LPs) concat. MQM	**0.578**	0.547	**0.278**	**0.468**
Final Ensemble				
Ensemble 11x	0.568	0.556	0.223	0.449

Table 2: Results for sentence-level QE in terms of Spearman correlation for MQM.

MQM. Results for sentence-level MQM systems are shown in Table 2. The results show that the two main techniques used for adapting to MQM data, filtering DA data to the three MQM LPs and using MQM data for fine-tuning, improved Spearman correlations for all LPs over the pure DA baseline, for both sentence-level and multi-task systems. However, these techniques improved certain LPs more than others, so combining them together improved multilingual scores even further. Overall, we noticed that our results for MQM data have a high variance. To mitigate this, we concatenated the DA and MQM datasets together for a single fine-tuning, resulting in our best individual system on our internal test set. Due to these peculiarities in the MQM LPs, we decided to ensemble systems tuned on both DA and MQM data. Our final ensemble did not have as strong results as the individual systems on our internal test set, yet, it showed superior performance upon submission to codalab leader-board.

Word-level. For the word-level task we tuned models separately for the LPs that consisted of post-edit-derived word tags and the ones consisting of MQM-derived word tags; we report the Matthew's correlation coefficient (MCC) in Table 3. We experimented with multi-tasking by adding sentence-level supervision to the word-level task and found

	Post-edit						MQM			
Method	**en-cs**	**en-ja**	**en-mr**	**km-en**	**ps-en**	**avg.**	**en-de**	**en-ru**	**zh-en**	**avg.**
Baseline (Zerva et al., 2021)	0.272	0.154	0.326	0.427	0.348	0.305	0.176	0.177	0.065	0.139
InfoXLM as encoder										
Word-level	0.351	0.183	0.337	0.443	0.372	0.337	-	-	-	-
+ Sentence-level	0.410	0.230	0.368	0.436	0.369	0.363	0.294	0.256	0.399	0.316
+ LP prefix	0.371	0.202	0.391	0.512	0.411	0.377	0.259	0.440	0.211	0.303
+ APEQuest & QT21	0.414	0.245	0.372	0.494	0.389	0.383	0.246	0.382	0.209	0.279
+ tuned class-weights	0.389	0.218	0.421	0.499	0.391	0.384	0.285	0.404	0.172	0.287
DA (3LPs) + MQM	-	-	-	-	-	-	0.265	0.367	0.360	0.331
RemBERT as encoder										
Word + sentence-level	0.353	0.163	0.303	0.443	0.369	0.326	0.262	0.309	0.147	0.240
+ LP prefix	0.384	0.257	0.375	0.460	0.370	0.369	0.288	0.356	0.297	0.313
Ensemble “best-only”	0.414	0.245	0.421	0.512	0.411	0.401	0.300	0.382	0.360	0.347
Ensemble logits	**0.438**	**0.257**	**0.445**	**0.547**	**0.430**	**0.423**	**0.325**	0.443	0.296	0.355
Ensemble tags	0.432	0.253	0.429	0.537	0.423	0.415	0.313	**0.446**	**0.408**	**0.389**

Table 3: Results for word-level QE in terms of MCC for the post-edit and MQM LPs. Note that in each row, we use models trained separately on the MQM and non-MQM LPs.

that it boosts performance especially for the out-of-English translations. For the non-MQM LPs we used the HTER scores as sentence level targets as we found they lead to significantly higher correlations. We can also see that using the sentence-mix and the language prefix boosted the performance for all LPs, both in the MQM and post-edit originated LPs. Overall, the results show further improvements when we use the HTER scores of APEQuest and QT21 as additional pretraining data, but only for specific LPs. These findings merit further investigation, since the directionality of the LPs seems to have impacted our experiments. Finally, ensembling led to better results across all languages. Ensembling the logits led to better results for the post-edit originated LPs, while word-level ensembling helped more the MQM-originated LPs. Yet, in the submitted versions we found that the difference in performance between the three ensembling methods yielded similar results, with only 1-2% difference, while in the averaged multilingual versions these differences were even smaller, varying less than 0.1%.

4.2 Explainable QE

Since the explanations are given as continuous scores, they are evaluated against the ground-truth word-level labels in terms of the Area Under the Curve (AUC), Average Precision (AP), and Recall at Top-K (R@K) metrics only on the subset of translations that contain errors. Although R@K was considered the main metric for this task, we optimized internally for the average of all three metrics. The results are shown in Table 4.

Discussion. The results highlight several contrasts between explanations for DA and MQM data: (i) while RemBERT is useful as an encoder for DA data (outperforms InfoXLM in 3 out of 5 LPs), it is outperformed by InfoXLM for all MQM LPs; (ii) the Head Mix component improves performance for DA, but it does not impact significantly the scores for MQM; and (iii) the Sparse Head Mix generally outperforms the Soft Head Mix for DA, but the trend flips for MQM. On what comes to the explainability methods, the baseline method (Attn × Norm – scaling the attention weights by the ℓ_2-norm of value vectors), which obtained the best results in last year's Explainable QE shared task, is outperformed by our new method (Attn × GradNorm) for both DA and MQM data. Moreover, ensembling explanations from different heads brings further consistent improvements across the board for all LPs. For the zero-shot setting (*en-yo*), we build an ensemble of explanations by using the heads that were more common among the ensembles for all other LPs. This approach might be worth researching further, since it is possible to study the Head Mix coefficients to select good-performing attention heads.

5 Official Results

We present the official results of our submissions alongside the results from other competitors in Section B for all three tasks. For sentence-level, our submissions achieved the best results for 6/9 LPs. For word-level, we obtained the best results for 5/9 LPs. For the explainable QE track, we obtained the

	Direct Assessment						MQM			
Method	**en-cs**	**en-ja**	**en-mr**	**km-en**	**ps-en**	**avg.**	**en-de**	**en-ru**	**zh-en**	**avg.**
Baseline (Treviso et al., 2021)[†]	0.602	0.510	0.428	0.636	0.633	0.562	0.529	0.552	0.450	0.510
InfoXLM as encoder										
Attn × GradNorm	0.602	0.495	0.417	0.653	0.648	0.563	0.539	0.559	0.474	0.524
+ Soft Head Mix	0.600	0.495	0.426	0.656	0.653	0.566	0.532	0.563	0.467	0.521
+ Sparse Head Mix	0.604	0.503	0.421	0.658	0.660	0.569	0.541	0.551	0.454	0.515
Ensemble	0.641	0.521	0.440	0.669	0.667	0.588	**0.580**	**0.603**	**0.505**	**0.563**
+ Soft Head Mix	0.621	0.501	0.432	0.681	0.661	0.579	0.567	0.588	0.504	0.553
+ Sparse Head Mix	**0.645**	0.519	**0.450**	0.688	0.675	0.595	0.574	0.582	0.484	0.547
RemBERT as encoder										
Attn × GradNorm	0.596	0.511	0.427	0.675	0.676	0.577	0.474	0.532	0.448	0.485
+ Soft Head Mix	0.588	0.538	0.430	0.658	0.654	0.574	0.473	0.529	0.455	0.486
+ Sparse Head Mix	0.588	0.534	0.428	0.658	0.652	0.572	0.470	0.530	0.443	0.481
Ensemble	0.609	0.551	0.443	**0.702**	0.685	0.598	0.516	0.554	0.506	0.525
+ Soft Head Mix	0.613	**0.561**	0.448	0.699	0.692	0.603	0.521	0.558	0.498	0.526
+ Sparse Head Mix	0.620	0.557	0.447	**0.702**	**0.691**	**0.604**	0.511	0.551	0.503	0.522

Table 4: Explainable QE task results in terms of the average of AUC, AP and R@K. [†]We used InfoXLM to compute the results for the baseline.

best results for all but two LPs (*km-en* and ps-en). Although the critical error detection task had no other competitor for the *constrained setting*, our submission vastly surpassed the organizers' baseline. We also obtained the best results for the multilingual settings (including and excluding *en-yo*) for all tasks. Finally, when averaging the results for all LPs, our submissions place on top for all tasks.

6 Conclusions and Future Work

We presented the joint contribution of IST and Unbabel to the WMT 2022 QE shared task. We found that incorporating references during pretraining improves performance across several LPs on downstream tasks, and that jointly training with sentence and word-level objectives yields a further boost. For Task 1, our final submissions were ensembles of models finetuned with different pretrained language models as encoders, boosting the results when compared to the previous year submission. For Task 2, we take inspiration on the literature of explainability and propose to use gradient information in tandem with attention weights, and to further refine the impact of attention heads towards the prediction via the Head Mix component. Besides leading to better explainability performance for some LPs, this strategy is potentially useful to identify good attention heads at inference time for zero-shot LPs, and deserves more investigation. Overall, our submissions achieved the best results for all tasks (including Task 3) for almost all LPs by a considerable margin.

One of the challenges of leveraging big ensembles is the burdensome weight of parameters and inference time. For future work we will extend our recent work, COMETINHO (Rei et al., 2022) and explore how to effectively distill large ensembles into small and more practical QE systems.

Acknowledgements

This work was supported by the P2020 programs (MAIA, contract 045909), by EU's Horizon Europe Research and Innovation Actions (UTTER, contract 101070631), by the European Research Council (ERC StG DeepSPIN, 758969), and by the Fundação para a Ciência e Tecnologia (contract UIDB/50008/2020).

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. Explaining predictions of non-linear classifiers in NLP. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 1–7, Berlin, Germany. Association for Computational Linguistics.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. Association for Computational Linguistics.

George Chrysostomou and Nikolaos Aletras. 2022. An empirical study on explanations in out-of-domain settings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6920–6938, Dublin, Ireland. Association for Computational Linguistics.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking Embedding Coupling in Pre-trained Language Models. In *International Conference on Learning Representations*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Marina Fomicheva, Piyawat Lertvittayakumjorn, Wei Zhao, Steffen Eger, and Yang Gao. 2021. The Eval4NLP shared task on explainable quality estimation: Overview and results. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 165–178, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Erick Fonseca, Frédéric Blain, Vishrav Chaudhary, Francisco Guzmán, Nina Lopatina, Lucia Specia, and André F. T. Martins. 2022. MLQE-PE: A Multilingual Quality Estimation and Post-Editing Dataset. In *Proceedings of the Language Resources and Evaluation Conference*, pages 4963–4974, Marseille, France. European Language Resources Association.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021a. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021b. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Nuno M. Guerreiro and André F. T. Martins. 2021. SPECTRA: Sparse structured text rationalization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6534–6550, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Julia Ive, Lucia Specia, Sara Szoc, Tom Vanallemeersch, Joachim Van den Bogaert, Eduardo Farah, Christine Maroti, Artur Ventura, and Maxim Khalilov. 2020. A post-editing dataset in the legal domain: Do we underestimate neural machine translation quality? In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3692–3697, Marseille, France. European Language Resources Association.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas. Association for Computational Linguistics.

Andre Martins and Ramon Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*, pages 1614–1623.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. TransQuest at WMT2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1049–1055, Online. Association for Computational Linguistics.

Ricardo Rei, Ana C Farinha, José G.C. de Souza, Pedro G. Ramos, André F.T. Martins, Luisa Coheur, and Alon Lavie. 2022. Searching for COMETINHO: The little metric that could. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 61–70, Ghent, Belgium. European Association for Machine Translation.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should I trust you?: Explaining the predictions of any classifier. In *Proc. ACM SIGKDD*, pages 1135–1144. ACM.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *In Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Lucia Specia, Kim Harris, Frédéric Blain, Aljoscha Burchardt, Viviven Macketanz, Inguna Skadin, Matteo Negri, and Marco Turchi. 2017. Translation quality and productivity: A study on rich morphology languages. In *Proceedings of Machine Translation Summit XVI: Research Track*, pages 55–71, Nagoya Japan.

Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold. 2018. Quality estimation for machine translation. *Synthesis Lectures on Human Language Technologies*, 11(1):1–162.

Marcos Treviso, Nuno M. Guerreiro, Ricardo Rei, and André F. T. Martins. 2021. IST-unbabel 2021 submission for the explainable quality estimation shared task. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 133–145, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pages 5998–6008. Curran Associates, Inc.

Yu Wan, Dayiheng Liu, Baosong Yang, Haibo Zhang, Boxing Chen, Derek Wong, and Lidia Chao. 2022. UniTE: Unified translation evaluation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8117–8127, Dublin, Ireland. Association for Computational Linguistics.

Chrysoula Zerva, Frédéric Blain, Ricardo Rei, Piyawat Lertvittayakumjorn, José G. C. de Souza, Steffen Eger, Diptesh Kanojia, Duarte Alves, Constantin Orăsan, Marina Fomicheva, André F. T. Martins, and Lucia Specia. 2022. Findings of the WMT 2022 Shared Task on Quality Estimation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi. Association for Computational Linguistics.

Chrysoula Zerva, Daan van Stigt, Ricardo Rei, Ana C Farinha, Pedro Ramos, José G. C. de Souza, Taisiya Glushkova, Miguel Vera, Fabio Kepler, and André F. T. Martins. 2021. IST-unbabel 2021 submission for the quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 961–972, Online. Association for Computational Linguistics.

A Data Information

The data used for finetuning our QE systems is shown in Table 5. For DA data, we split the original development set to generate a new dev/test split, therefore the reported numbers in the table correspond to this "internal" dev split.

LP	Samples	Source Tokens	Target Tokens	Target OK / BAD
TRAIN				
en-de	9000	147870	153656	0.84 / 0.16
en-mr	26000	690516	561371	0.90 / 0.10
en-zh	9000	148657	163308	0.65 / 0.35
et-en	9000	126877	185491	0.75 / 0.25
ne-en	9000	135205	181707	0.41 / 0.59
ro-en	9000	154538	167471	0.71 / 0.29
ru-en	9000	104423	132006	0.85 / 0.15
si-en	9000	141283	166914	0.42 / 0.58
en-de†	54681	1571090	1926444	0.90 / 0.10
en-ru†	15628	312185	354871	0.95 / 0.05
zh-en†	75327	134165	2789907	0.87 / 0.13
DEV				
en-de	500	8262	8555	0.84 / 0.16
en-mr	500	13803	11216	0.91 / 0.09
en-zh	500	8422	9302	0.75 / 0.25
et-en	500	7081	10257	0.73 / 0.27
ne-en	500	7542	10247	0.38 / 0.62
ro-en	500	8550	9202	0.78 / 0.22
ru-en	500	5984	7511	0.84 / 0.16
si-en	500	7866	9415	0.41 / 0.59
en-cs	500	10302	9302	0.75 / 0.25
en-ja	500	10354	13287	0.73 / 0.27
km-en	495	9015	8843	0.45 / 0.55
ps-en	500	13463	12160	0.51 / 0.49
en-de†	503	10535	12454	0.96 / 0.04
en-ru†	503	10767	11911	0.91 / 0.09
zh-en†	509	980	19192	0.98 / 0.02

Table 5: DA and MQM (†) data for all LPs.

B Official Results

Critical Error Detection. Submissions for this task were evaluated in terms of ranking using R@K and MCC as metrics. In Table 7, we report only MCC scores as it was the main metric for this task.

QE and Explainable QE. Table 6 shows the official results for sentence-level QE (top), word-level QE (middle), and explainable QE (bottom).

	Direct Assessment								**MQM**		
Team	**en-cs**	**en-ja**	**en-mr**	**en-yo**	**km-en**	**ps-en**	**all**	**all/yo**	**en-ru**	**en-de**	**zh-en**
Sentence-level QE											
Baseline	0.560	0.272	0.436	0.002	0.579	0.641	0.415	0.497	0.333	0.455	0.164
Alibaba	-	-	-	-	-	-	-	-	0.505	0.550	0.347
NJUQE	-	-	0.585	-	-	-	-	-	0.474	**0.635**	0.296
Welocalize	0.563	0.276	0.444	-	0.623	-	0.448	0.506	-	-	-
joanne.wjy	0.635	0.348	0.597	-	0.657	0.697	-	0.587	-	-	-
HW-TSC	0.626	0.341	0.567	-	0.509	0.661	-	-	0.433	0.494	**0.369**
Papago	0.636	0.327	**0.604**	0.121	0.653	0.671	0.502	0.571	0.496	0.582	0.325
IST-Unbabel	**0.655**	**0.385**	0.592	**0.409**	**0.669**	**0.722**	**0.572**	**0.605**	**0.519**	0.561	0.348
Word-level QE											
Baseline	0.325	0.175	0.306	0.000	0.402	0.359	0.235	0.257	0.203	0.182	0.104
NJUQE	-	-	0.412	-	0.421	-	-	-	0.390	**0.352**	0.308
HW-TSC	0.424	**0.258**	0.351	-	0.353	0.358	-	0.218	0.343	0.274	0.246
Papago	0.396	0.257	**0.418**	0.028	**0.429**	0.374	0.317	0.343	0.421	0.319	0.351
IST-Unbabel	**0.436**	0.238	0.392	**0.131**	0.425	**0.424**	**0.341**	**0.361**	**0.427**	0.303	**0.360**
Explainable QE											
Baseline	0.417	0.367	0.194	0.111	0.580	0.615	0.381	0.435	0.148	0.074	0.048
f.azadi	-	-	-	-	0.622	0.668	-	-	-	-	-
HW-TSC	0.536	0.462	0.280	-	**0.686**	**0.715**	-	0.535	0.313	0.252	0.220
IST-Unbabel	**0.561**	**0.466**	**0.317**	**0.234**	0.665	0.672	**0.486**	**0.536**	**0.390**	**0.365**	**0.379**

Table 6: Official results for sentence-level QE (top) in terms of Spearman's correlation, word-level QE (middle) in terms of MCC, and explainable QE (bottom) in terms of R@K. We estimated the numbers of *en-yo* for teams that did not submit to *en-yo* directly but still submitted to all other LPs and to the *multilingual* (all) category.

Method	**en-de**	**pt-en**
Baseline	0.0738	-0.0013
InfoXLM finetuned on DAs	0.5641	0.7209

Table 7: Official results for the Critical Error Detection task in terms of MCC.

CrossQE: HW-TSC 2022 Submission for the Quality Estimation Shared Task

Shimin Tao[1*], **Chang Su**[1*], **Miaomiao Ma**[1], **Hao Yang**[1], **Min Zhang**[1],
Xiang Geng[2], Shujian Huang[2], Jiaxin Guo[1], Minghan Wang[1], Yinglu Li[1]
[1]Huawei Translation Services Center, China
[2]Nanjing University, China
{taoshimin, suchang8, mamiaomiao, yanghao30, zhangmin186, guojiaxin1,
wangminghan, liyinglu}@huawei.com
gx@smail.nju.edu.cn, huangsj@nju.edu.cn

Abstract

Quality estimation (QE) is a crucial method to investigate automatic methods for estimating the quality of machine translation results without reference translations. This paper presents Huawei Translation Services Center's (HW-TSC's) work called CrossQE in WMT 2022 QE shared tasks 1 and 2, namely sentence- and word- level quality prediction and explainable QE. CrossQE employes the framework of predictor-estimator for task 1, concretely with a pre-trained cross-lingual XLM-RoBERTa large as predictor and task-specific classifier or regressor as estimator. An extensive set of experimental results show that after adding bottleneck adapter layer, mean teacher loss, masked language modeling task loss and MC dropout methods in CrossQE, the performance has improved to a certain extent. For task 2, CrossQE calculated the cosine similarity between each word feature in the target and each word feature in the source by task 1 sentence-level QE system's predictor, and used the inverse value of maximum similarity between each word in the target and the source as the word translation error risk value. Moreover, CrossQE has outstanding performance on QE test sets of WMT 2022.

1 Introduction

Quality estimation (QE) is the task of evaluating a translation system's quality without access to reference translations (Specia et al., 2018). In WMT 2022 QE shared task [1], there are three tasks — Quality Prediction, Explainable QE and Critical Error Detection. Each task involves several language pairs. Our team — Huawei Translation Services Center (HW-TSC) — participated in quality prediction and explainable QE tasks over all language pairs.

This paper describes the HW-TSC's systems called CrossQE submitted for these tasks. Some key steps are summarized as follow:

- We used pre-trained Cross-lingual XLM-Roberta large (Lample and Conneau, 2019; Conneau et al., 2019) as predictor instead of RNN-based model in the two-stage Predictor-Estimator architecture (Kim et al., 2017). The task-specific classifier or regressor is used as quality estimator, and multitasks are trained at the same time.

- The cross-lingual XLM-RoBERTa large model is pre-trained on large-scale parallel corpora where source and target tokens are concatenated by MLM task. Shuffling those tokens and predicting those tokens' index by the pre-trained model as an additional pre-training task can improve the QE model's effect.

- We build on the COMET architecture [2] by exploring adapter layers (Houlsby et al., 2019) for quality estimation to eliminate the overfitting problem instead of fine-tuning the whole base pre-trained model for different NLP tasks (He et al., 2021).

- In the training step, the Mean Teacher loss (Baek et al., 2021) was added to improve model's over-fitting problem.

- We explored data augmentation a method based on Monte Carlo (MC) dropout (Gal and Ghahramani, 2016) which to enhance the performance in sentence-level Direct Assessment (DA) and Multidimentional Quality Metrics (MQM) score task. During prediction, the dropout function is still enabled, and the prediction is performed for N times. The average value of the prediction is the final prediction value.

[*] Indicates equal contribution.
[1] https://wmt-qe-task.github.io/
[2] https://github.com/Unbabel/COMET

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 646 - 652
December 7-8, 2022 ©2022 Association for Computational Linguistics

- We used QE model's predictor of the sentence-level quality prediction sub-task as a words' features. extractor cosine similarity's opposite value of target words' vectors extracted from the predictor trained from sentence-level quality prediction sub-task between source and target as the explainable QE task's token-level scores.

Our methods achieve impressive performance on both sentence- and word- level tasks. Specifically, we peak the top-1 on quality prediction sentence-level sub-task over Chinese-English language pair and word-level sub-task over English-Japanese language pair. We also win the first place in explainable QE task in Khmer-English and Pashto-English language pairs. We will describe the datasets and our methods for those tasks in section 2 and section 3. Section 4 presents details of our experimental setup and results. In section 5, a brief discussion and conclusion are presented.

2 Task & Data Set

2.1 Task Description

Task 1

The quality prediction task follows the trend of the previous years in comprising a sentence-level sub-task where the goal is to predict the quality score for each source-target sentence pair and a word-level sub-task where the goal is to predict the translation errors, assigning OK/BAD tags to each word of the target. Both sub-tasks include annotations derived in two different ways, depending on the language pair: direct assessment (DA), following the trend of the previous years, and multidimensional quality metrics (MQM), introduced for the first time in the QE shared task. The sentence- and word-level sub-tasks use the same source-target sentences for each language pair.

Task 2

The explainable QE task proposes to address translation error identification as rationale extraction. Instead of training a dedicated word-level model, to infer translation errors as an explanation for sentence-level quality scores, a list of continuous token-level scores where the tokens with the highest scores are expected to correspond to translation errors should be calculated.

2.2 Data Set & Data Processing

Some information about the data set is as follow:

There are three language pairs annotated with MQM annotations for training/development/test set: English-Russian (En-Ru), English-German (En-De), Chinese-English (Zh-En) and the one language pair annotated with DA annotations for training/development/test set: English-Marathi (En-Mr).

The data set of these four language pairs contains 15k training data for En-Ru, 26k training data for En-De, 31k training data for Zh-En, 26k training data for En-Mr and 1k development data for each language pair.

There are seven language pairs annotated with DA annotations for training/development ser: English-German (En-De), English-Chinese (En-Zh), Esthonian-English (Et-En), Nepali-English (Ne-En), Romanian-English (Ro-En), Russian-English (Ru-En), Sinhala-English (Si-En), and four zero-shot language pairs annotated with DA annotations for test set: English-Czech (En-Cs), English-Japanese (En-Ja), Khmer-English (Km-En) and Pashto-English (Ps-En). The data set of these seven language pairs contains 9k training data and 1k development data.

The word-level sub-task data set consists of predicting word-level tags for the target side (to detect mistranslated or missing words). Each token is tagged as either OK or BAD. The OK/BAD tags are provided for each of the language pairs of the sentence-level task, and are derived from either MQM annotations (En-De, Zh-En and En-Ru) or post-edited sentences.

So for MQM language pairs, it is a few-shot task, and for DA language pairs, it is a zeor-shot task. For training data of each language pair, sentence scores are linearly normalized from 0 to 1, and can be restored to the original value, so a multilingual sentence-level QE model can be trained for all language pairs.

3 Methodology

3.1 System

Task 1

Our quality estimator system follows the two-stage Predictor-Estimator architecture, which uses a languange model encoder as predictor and using task-specific classifier or regressor as estimator (Chen et al., 2021). In our system, the predictor is a pre-trained cross-lingual XLM-RoBERTa model (f). For the sentence-level quality score prediction task, the estimator is a regressor (σ_{score}), and for the word-level quality label prediction task, the estimator is a classifier (σ_{class}), as depicted in figure 1.

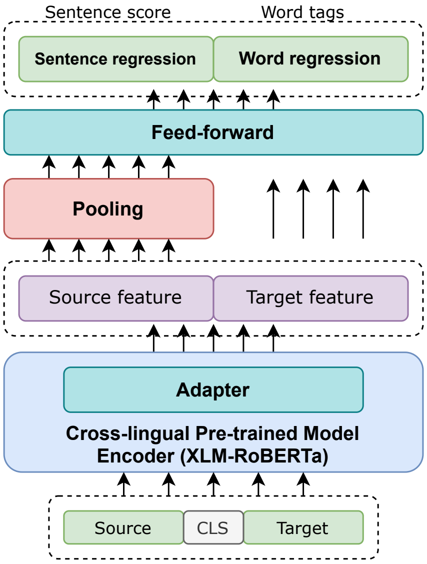


Figure 1: The architecture of CrossQE quality estimator system

Sentence-level

After the predictor obtaining tokens embedding features (H_s, H_t; where H_s for source embedding features and H_s for target embedding features), we use masked pooling p to calculate the entire source or target sentence feature vector. In the experiment, we put combination of [source, target] ($[S, T]$) and [target, source] ($[T, S]$) as the input data into the predictor, and get four types of sentence feature vectors ($F_{s_A}, F_{t_A}, F_{t_B}, F_{s_B}$). All the sentence feature vectors are combined to the estimator perform score prediction, and the performance is improved obviously.

Word-level

In the task, OK is set to 1 and BAD is set to 0, thus the word-level estimator becomes a binary classification model. To avoid overfitting, the OK label is set to 0.9, the BAD label is set to 0.1, and the index 0's value of outputs softmax logits is used as the word quality score ($V_{w-score}$). The mean squared error (MSE) loss is calculated on the outputs and labels and the word-level QE model is updated. In the prediction phase, if the output word score is greater than 0.5, it is considered as an OK label. Otherwise, it is considered as a BAD label.

The equation of task 1 is shown as equation 1:

$$\begin{aligned}
H_{s_A}, H_{t_A} &= f([S, T]), \\
F_{s_A}, F_{t_A} &= p(H_{s_A}, H_{t_A}), \\
H_{t_B}, H_{s_B} &= f([T, S]), \\
F_{t_B}, F_{s_B} &= p(H_{t_B}, H_{s_B}), \\
V_{score} &= \sigma_{score}([F_{s_A}, F_{t_A}, F_{s_B}, F_{t_B}]), \\
V_{class} &= \sigma_{class}([H_{s_B}, H_{t_B}]), \\
V_{w-score} &= softmax(V_{class})[0]
\end{aligned} \tag{1}$$

Where V_{score} is output of the sentence-level estimator and V_{class} is logits of the word-level estimator.

Task 2

We use the sentence-level QE model's predictor from task 1 as a sentence word embedding feature extractor. Similarity is used as the possibility of word translation (Yang et al., 2022). If a word in target is highly similar to a word in source, the word translation is correct. Otherwise, the word translation is incorrect. The higher the similarity, the higher the probability of correct translation, and vice versa.

We extracted the source and target sentence embedding features by word and calculated the cosine similarity between each word feature in the target and each word feature in the source. The maximum similarity between each word in the target and the source is used as the score of the word translation quality. We used the inverse of the quality score of each word in the target as the translation error risk value, so each target sentence can obtain a word error risk value list, in which a higher score indicates a higher probability of incorrect translation.

3.2 Model Pre-training

Cross-lingual Language Model

As XLM-RoBERTa, a multilingual model that can override the QE tasks' language pairs, does a good job with language tasks, it was chosen as the predictor. Cross-lingual language model pre-training is outstanding in low-resource training data. We add [CLS] between the tokens of the source text and the tokens of the target text and input the combined tokens to the XLM-RoBERTa model for masked language modeling (MLM) task pre-training (Devlin et al., 2018) to enhance the model's ability to understand words and sentences between languages. We sampled randomly 15% of the sub-word tokens from the text streams, replaced them by a [MASK] token in 80% probability, by a random token in 10% chance, and we keeped them unchanged in 10% chance.

Token Shuffling Pre-training

We randomized the sequence of input tokens and let the cross-lingual language model predict the sequence number of' each token. This pre-training task has an obvious positive effect on word-level QE sub-task. Because the model has never done a position prediction task, the training task is divided into two stages for the sake of training stability. In stage one, 50% of the tokens are selected and shuffled, and in stage two, all the tokens are shuffled.

3.3 Bottleneck Adapter Layer

The provided training set is relatively small, making the model to be easily over-fitted if all weights are updated. Therefore, we decided to integrate the Bottleneck Adapter Layers (BAL) (Wang et al., 2020) while keeping parameters of the original Transformer fixed (Yang et al., 2020). The bottle with a "thick" neck could further improve the performance without seriously sacrificing training efficiency. By increasing the parameter size of BALs, the performance also increased linearly, finally reaching the peak of 104% of the baseline performance with the neck having twice the hidden size.

3.4 Model Training

Mean Teacher Loss

Mean teacher is a method that uses consistency regularization. As shown in figure 2, the process is as follows:

1) Copy the predictor as a teacher model and the original model as a student model.

2) At the training step, apply two random augmentations η and η' on the same mini-batch tokens' embedding features.

3) Input the former data $(Input_{embedding} + \eta)$ to the student model and the latter data $(Input_{embedding} + \eta')$ to the teacher model.

4) Calculate the MSE loss on their outputs.

5) Use the MSE loss to update the t th iter's parameters of the student model $P_{stu}[t]$.

6) Use the exponential moving average (EMA) method to update the t th iter's parameters of the teacher model $P_{tea}[t]$ as shown in equation 2.

$$P_{tea}[t] = \alpha \times P_{tea}[t-1] + (1-\alpha) \times P_{stu}[t] \quad (2)$$

Where, α is a hyperparameter (0.95 in this paper).

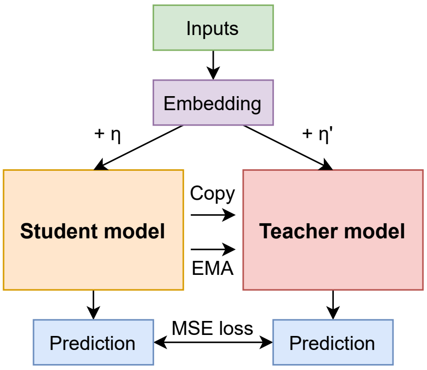


Figure 2: Mean teacher loss

MLM Task Loss

To further improve the language understanding capability of the model, we add MLM task loss into the training. We find adding MLM training task during the training of sentence-level and word-level QE models for multi-task training can improve the model performance.

Total loss for the CrossQE system to update model parameters is shown as equation 3:

$$\begin{aligned} Loss = \alpha_1 \times [Loss_s|Loss_w] + \\ \alpha_2 \times Loss_{MT} + \alpha_3 \times Loss_{MLM} \end{aligned} \quad (3)$$

Where, α_1, α_2 and α_3 are hyperparameters, $[Loss_s|Loss_w]$ is the sentence-level or word-level sub-task training loss, $Loss_{MT}$ is the mean teacher loss and $Loss_{MLM}$ is the MLM task loss.

4 Experiments & Results

4.1 Model Settings

We followed the model settings of COMET (Rei et al., 2022) to fine-tune our QE model based on the XLM-RoBERTa large model [3] with a classification/regression head on a single V100 GPU. The XLM-RoBERTa large model pre-trained on 2.5TB of filtered CommonCrawl data containing 100 languages is a multilingual version of RoBERTa which is a transformers model pretrained on a large corpus in a self-supervised fashion. It has approximately 550M parameters and 24 hidden encode layers. The training batch size is set to 4, the gradient accumulation number is set to 4 and it took about 2 hours for the model to converge in the training step. The XLM-RoBERTa large model has been pre-trained on the WMT 2021 news translation shared task's parallel corpora [4] by model pre-training methods described in the section 3.2.

[3] https://huggingface.co/xlm-roberta-large

[4] https://www.statmt.org/wmt21/translation-task.html

model	Language									
	En-Ru	En-De	Zh-En	En-Mr	En-Zh	Et-En	Ne-En	Ro-En	Ru-En	Si-En
baseline	0.3852	0.4436	0.3148	0.5123	0.2437	0.4635	0.5379	0.3572	0.4699	0.6109
M-Cross	0.4403	0.4807	0.3796	0.5419	0.2911	0.4827	0.5744	0.3899	0.4712	0.6358
M-Adapter	0.4487	0.4926	0.3815	0.5547	0.2938	0.4913	0.5899	0.4003	0.4962	0.6471
M-MT	0.4531	0.4917	0.3827	0.5681	0.3094	0.5083	0.6092	0.4090	0.5101	0.6566
M-MLM	0.4599	0.4928	0.3812	0.5679	0.3008	0.5101	0.6044	0.4182	0.5062	0.6653
M-Final	**0.4730**	**0.5228**	**0.4002**	**0.5937**	**0.3247**	**0.5336**	**0.6217**	**0.4483**	**0.5211**	**0.6973**

Table 1: Results of the task 1 sentence-level's spearman coefficient performance on development set over ten language pairs.

model	Language									
	En-Ru	En-De	Zh-En	En-Mr	En-Zh	Et-En	Ne-En	Ro-En	Ru-En	Si-En
baseline	0.3182	0.2777	0.2643	0.3655	0.4007	0.2653	0.4432	0.3705	0.3642	0.4201
M-Adapter	0.3248	0.2796	0.2711	0.3681	0.4052	0.2714	0.4506	0.3832	0.3795	0.4588
M-MT	0.3274	0.3003	0.2807	0.3617	0.4201	0.2885	0.4494	0.3997	0.3814	0.4473
M-Final	**0.3671**	**0.3112**	**0.2997**	**0.3872**	**0.4447**	**0.2963**	**0.4704**	**0.4041**	**0.3894**	**0.4960**

Table 2: Results of the task 1 word-level's target words' MCC performance on development set over ten language pairs.

4.2 Experiments of Sentence-level QE Task

In our experiment, we set $\alpha_1 = 1.0$, $\alpha_2 = 0.5$ and $\alpha_3 = 0.5$ (we also set $\alpha_1 = 1.0$, $\alpha_2 = 1.0$, $\alpha_3 = 1.0$ or $\alpha_1 = 0.5$, $\alpha_2 = 1.0$, $\alpha_3 = 0.5$ or $\alpha_1 = 0.5$, $\alpha_2 = 0.5$, $\alpha_3 = 1.0$ or $\alpha_1 = 0.5$, $\alpha_2 = 1.0$, $\alpha_3 = 1.0$, but all of them can not get the best result). Our baseline model is the COMET's open-source framework model with the self pretrained XLM-RoBERTa model as predictor. The primary evaluation metric for the sentence-level sub-task of Task 1 is the spearman r coefficient as show in Table 1.

Obviously, the performance of the baseline model is relatively poor. By leveraging Cross-lingual language model as predictor (M-Cross model), the model achieved much better performance. Adding the BAL adapter (M-Adapter model) into Cross-lingual language model, the effect is further improved. In the experiment, it is found that excessive training leads to reduced effectiveness on development set, while the addition of mean tearcher loss (M-MT model) can significantly suppress the overfitting problem and further improve the model performance. Adding the MLM loss (M-MLM model) to the training process enhances the model performance to some degree. Finally, the MC dropout method is used to predict the QE sentence-level scores (M-Final model), which can improve the performance coefficient by at least 1%.

Language	Spearman
En-Ru	0.4329
En-De	0.4939
Zh-En	0.3685
En-Mr	0.5672
En-Cs	0.6257
En-Ja	0.3409
Km-En	0.5087
Ps-En	0.6608

Table 3: Results of the task 1 sentence-level's spearman coefficient performance on the test set over eight language pairs.

Finally, we committed our results of M-Final model on the test set. The performance of the system on the test set is shown in Table 3. For the zero-shot data, the system also has good performance. Specifically, we get the best performance on Zh-En language pair.

4.3 Experiments of Word-level QE Task

In our experiment, we set $\alpha_1 = 0.5$, $\alpha_2 = 1.0$ and $\alpha_3 = 1.0$ (we also set $\alpha_1 = 1.0$, $\alpha_2 = 1.0$, $\alpha_3 = 1.0$ or $\alpha_1 = 0.5$, $\alpha_2 = 1.0$, $\alpha_3 = 0.5$ or $\alpha_1 = 0.5$, $\alpha_2 = 0.5$, $\alpha_3 = 1.0$ or $\alpha_1 = 1.0$, $\alpha_2 = 0.5$, $\alpha_3 = 0.5$, but all of them can not get the best result). Our baseline model is the cross-lingual language model that is used as predictor by the COMET's open-source framework. The primary evaluation

Language	MCC
En-Ru	0.3425
En-De	0.2739
Zh-En	0.2457
En-Mr	0.3509
En-Cs	0.4239
En-Ja	0.2576
Km-En	0.3531
Ps-En	0.3576

Table 4: Results of the task 1 word-level's target words' MCC performance on the test set over eight language pairs.

Language	Recall
En-Ru	0.3126
En-De	0.2517
Zh-En	0.2203
En-Mr	0.2800
En-Cs	0.5356
En-Ja	0.4617
Km-En	0.6863
Ps-En	0.7151

Table 5: Results for the task 2 target recall at top-K's performance on the test set over eight language pairs.

metric for the word-level sub-task of Task 1 is the matthews correlation coefficient (MCC) as shown in Table 1.

Compared with the baseline, the model has better performance after the BAL adapter is added (M-Adapter model). Also, the addition of mean tearcher loss (M-MT model) can further improve the model pereformance. However, we found after adding the MLM loss to the training process (M-Final model), there were no significant improvement in pereformance.

Finally, we committed our results of M-Final model on the test set. The performance of the system on the test set is shown in Table 4. For the zero-shot data, the system also has good performance. Specifically, we get the best performance on En-Ja language pair.

4.4 Experiments of Explainable QE Task

As stated in the mission requirements, the participants are not allowed to supervise their models with any token-level or word-level labels or signals (whether they are from natural or synthetic data) in order to directly predict word-level errors. We just used the sentence-level quality prediction model's predictor as the sentence word embedding feature extractor, and calculated the translation error risk value as stated in section 3.1.

Finally, we committed our results on the test set. The performance of the system on the test set is shown in the Table 5. We get the best performance on the Km-En and Ps-En language pairs.

5 Conclusion

This paper presents HW-TSC's work called CrossQE on WMT 2022 QE shared task. CrossQE got the first place in four single projects. For the tasks 1, CrossQE employed the predictor-estimator framework as baseline. To further boost performance, we investigated the usage of pre-trained cross-lingual XLM-RoBERTa large language model as predictor, and added the bottleneck adapter layer into the predictor to mitigate overfitting issues. For both sentence- and word- level sub-task, we added mean teacher loss and MLM task loss into model training step, and added MC dropout at the inference step in sentence-level sub-task. Those methods delivered a good performance in all language pairs, including zero-shot language pairs. For task 2, we used the sentence-level QE model's predictor from task 1 as a sentence word embedding feature extractor, and used the inverse value of maximum similarity between each word in the target and the source as the word translation error risk value. In future, we will invest time and effort in studying the effect of involving additional translations into QE tasks, for example, how the additional translation quality will affect QE performance.

References

Jeonghun Baek, Yusuke Matsui, and Kiyoharu Aizawa. 2021. What if we only use real datasets for scene text recognition? toward scene text recognition with fewer labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3113–3122.

Yimeng Chen, Chang Su, Yingtao Zhang, Yuxia Wang, Xiang Geng, Hao Yang, Shimin Tao, Guo Jiaxin, Wang Minghan, Min Zhang, et al. 2021. Hw-tsc's participation at wmt 2021 quality estimation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 890–896.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco

Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jia-Wei Low, Lidong Bing, and Luo Si. 2021. On the effectiveness of adapter-based tuning for pretrained language model adaptation. *arXiv preprint arXiv:2106.03164*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Hyun Kim, Hun-Young Jung, Hongseok Kwon, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator: Neural quality estimation based on target word prediction for machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 17:1–22.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Ricardo Rei, Ana C Farinha, José G.C. de Souza, Pedro G. Ramos, André F.T. Martins, Luisa Coheur, and Alon Lavie. 2022. Searching for COMETINHO: The little metric that could. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 61–70, Ghent, Belgium. European Association for Machine Translation.

Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold. 2018. Quality estimation for machine translation. *Synthesis Lectures on Human Language Technologies*, 11(1):1–162.

Minghan Wang, Hao Yang, Hengchao Shang, Daimeng Wei, Jiaxin Guo, Lizhi Lei, Ying Qin, Shimin Tao, Shiliang Sun, Yimeng Chen, et al. 2020. Hw-tsc's participation at wmt 2020 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1056–1061.

Hao Yang, Minghan Wang, Ning Xie, Ying Qin, and Yao Deng. 2020. Efficient transfer learning for quality estimation with bottleneck adapter layer. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 29–34.

Hao Yang, Min Zhang, Shimin Tao, Miaomiao Ma, Ying Qin, and Chang Su. 2022. TeacherSim: Cross-lingual machine translation evaluation with monolingual embedding as teacher. In *The 2nd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*. To be publiushed.

Welocalize-ARC/NKUA's Submission to the WMT 2022 Quality Estimation Shared Task

Eirini Zafeiridou[1,2], **Sokratis Sofianopoulos**[3]
[1]Welocalize Inc / Frederick, MD, United States
[2]National and Kapodistrian University of Athens / Athens, Greece
[3]Institute for Language and Speech Processing - Athena RC / Athens, Greece
eirini.zafeiridou@welocalize.com, s_sofian@athenarc.gr

Abstract

This paper presents our submission to the WMT 2022 quality estimation shared task and more specifically to the quality prediction sentence-level direct assessment (DA) subtask. We build a multilingual system based on the predictor–estimator architecture by using the XLM-RoBERTa transformer for feature extraction and a regression head on top of the final model to estimate the z-standardized DA labels. Furthermore, we use pretrained models to extract useful knowledge that reflect various criteria of quality assessment and demonstrate good correlation with human judgements. We optimize the performance of our model by incorporating this information as additional external features in the input data and by applying Monte Carlo dropout during both training and inference.

1 Introduction

Machine translation quality estimation (MTQE) is the task of automatically estimating the quality of the MT output without using reference translations or any other human input (Blatz et al., 2004; Specia et al., 2009, 2018). MTQE has many use cases and can be applied in various settings (Specia and Shah, 2018). It can be used to estimate the post-editing effort, to rank and compare outputs of different MT systems or to classify the segments that need post-editing. It can also be used to estimate the quality of the final translations as well as to filter out noisy segments from translation memories or training datasets. MTQE techniques usually have multiple granularity levels and can be applied to a word, phrase, sentence or even to an entire document. Such systems are highly efficient when a vast amount of machine translated segments need to be evaluated in less time, with less effort and lower costs compared to traditional evaluation techniques.

The WMT 2022 quality estimation shared task includes the following separate tasks: quality prediction, explainable QE and critical error detection. Our team participated in the quality prediction sentence-level direct assessment (DA) subtask with a multilingual MTQE system.

Specifically, we developed a cross-lingual MTQE system following the predictor–estimator architecture (Kim and Lee, 2016; Kim et al., 2017). We used the large-scale pretrained XLM-RoBERTa (XLM-R)[1] model (Conneau et al., 2020) for feature extraction, similarly to Chen et al. (2021). We combined the model's output with additional external features that demonstrate good correlation with the target variable. We then used the concatenated vector as input to our final MTQE regression model. Our regressor is a feed-forward neural network with a linear output layer used to estimate the z-standardized DA labels.

2 Quality prediction: sentence-level direct assessment

The quality prediction task of the WMT 2022 quality estimation shared task consists of a sentence-level and a word-level subtask. Using the provided annotated training data, the objective of the sentence-level direct assessment subtask is to develop a system that automatically estimates a quality score for each provided sentence pair which is highly correlated with human-generated z-standardized DA values.

2.1 Data

According to the instructions, for each language pair, participants can use all the annotations offered for the quality estimation shared tasks of the preceding year(s) that are accessible through the MLQE-PE GitHub page.[2]

MLQE-PE is a multilingual dataset for quality estimation which includes 11 language combinations covering low, medium and high re-

[1] https://huggingface.co/xlm-roberta-large
[2] https://github.com/sheffieldnlp/mlqe-pe

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 653 - 660
December 7-8, 2022 ©2022 Association for Computational Linguistics

source languages (Fomicheva et al., 2020a,c). The dataset is mainly created by translating sentences from Wikipedia articles using cutting-edge transformer NMT models, and by having expert linguists annotate the translations based on a modified version of DA ratings. Each sentence is annotated individually using the FLORES setup (Guzmán et al., 2019), in which three qualified translators provide evaluations on a scale of 0–100 based on their perceived translation quality. Raw DA scores are then standardized and transformed into z-scores by using the mean and standard deviation of every single annotator. The z-standardized per-annotator values are then averaged in order to get one final score for every translation.

The organizers also provide additional train, development, and test sets for the English–Marathi language pair that is not included in the MLQE-PE dataset.

language pair	Train	Dev.	Test
en–mr	26000	1000	1000
en–cs	–	1000	1000
en–ja	–	1000	1000
km–en	–	1000	1000
ps–en	–	1000	1000
en–de	9000	1000	–
en–zh	9000	1000	–
et–en	9000	1000	–
ne–en	9000	1000	–
ro–en	9000	1000	–
ru–en	9000	1000	–
si–en	9000	1000	–
en–yo	–	–	1000
total	89000	12000	6000

Table 1: Size of the provided train, development and test sets per language (in sentences)

The data for the sentence-level quality prediction subtask can be downloaded from the task's GitHub page.[3] The number of the available sentences per language is illustrated in the Table 1.

According to the instructions, it is also feasible to use the DA annotations that were generated for the metrics shared tasks in previous years.

For the training of our models, we use only the training part of the data provided by the organizers. For the English–Japanese language pair, we also use the training data of the 2020 metrics shared task.

2.2 Evaluation

This year, the primary evaluation metric for the sentence-level DA subtask is the *Spearman*'s rank correlation coefficient which is used to reflect the correlation between the predicted scores and the human annotated z-standardized DA labels. Secondary metrics also include MAE, RMSE, and *Pearson*'s correlation coefficient.

3 Method

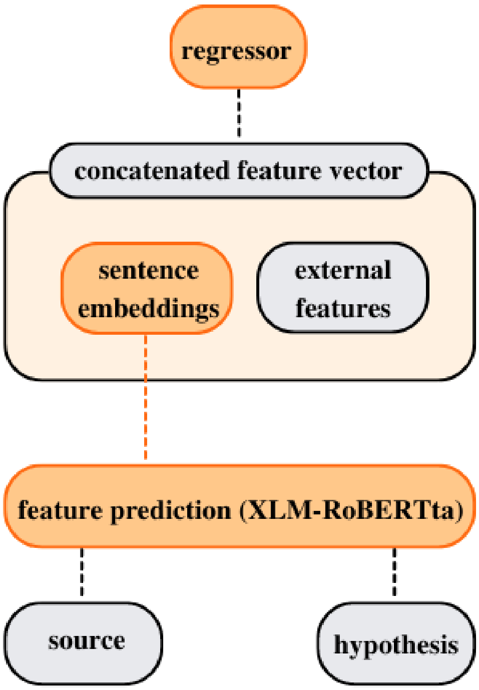


Figure 1: Model architecture

For the sentence-level direct assessment subtask we build and use a system based on the predictor–estimator architecture (Kim and Lee, 2016; Kim et al., 2017). Following similar state-of-the-art approaches (Fomicheva et al., 2020b; Moura et al., 2020; Rei et al., 2020; Zerva et al., 2021; Chen et al., 2021; Wang et al., 2021) we choose the pretrained XLM-RoBERTa[1] model (Conneau et al., 2020) to encode the input sequences and predict our features. We keep the XLM-R encoder frozen during training and we use it to generate cross-lingual representations over the source sentences and their corresponding translations. We then concatenate the output with additional external features and we feed the final feature vector to a feed-forward layer to finally estimate the continuous

[3] https://github.com/WMT-QE-Task/wmt-qe-2022-data/

z-standardized DA scores. We also employ Monte Carlo dropout during both training and inference to optimize the performance of our model. We use the mean-squared-error loss function and the AdamW (Loshchilov and Hutter, 2019) optimizer with a learning rate of 10^{-5}. The architecture of our model is illustrated in the Figure 1.

3.1 Cross-lingual Representations

In order to extract cross-lingual representations for each sentence pair, we start by encoding each source sentence and its hypothesis separately. From the output vectors, we extract the <s> classification token (equivalent to the [CLS]) that corresponds to the representation of the whole sequence (Ranasinghe et al., 2020). Then, similarly to the methodology proposed in RUSE (Shimanaka et al., 2018), we use the following sentence embeddings:

- Source embedding representation: $\vec{s}$
- Hypothesis embedding representation: $\vec{h}$
- Element-wise product: $\vec{s} \circ \vec{h}$
- Element-wise absolute difference: $| \vec{s} - \vec{h} |$

Motivated by the implementation of Rei et al. (2020), we concatenate the above representations into a single vector. Furthermore, we enrich the vector with additional external features $\vec{f}$ resulting in a final feature vector $\vec{x} = \left[\vec{f}; \vec{h}; \vec{s}; \vec{s} \circ \vec{h}; |\vec{s} - \vec{h}|\right]$, which is used as input to the output layer of our model.

3.2 Additional external features

Fomicheva et al. (2020c) suggested the use of glass-box features to predict the quality of the NMT outputs. Specifically, they proposed methods to quantify the model's uncertainty in unsupervised QE scenarios. Moura et al. (2020) and Zerva et al. (2021) also used such glass-box features as an effective strategy for the development of their QE systems. In our approach, we use the sentence-level NMT model scores included in the MLQE-PE dataset (Fomicheva et al., 2020a,c) and we further explore additional characteristics that can be effectively used in similar QE settings. We suggest a set of external features that reflect various criteria of translation quality assessment and exhibit good correlation with human judgements, as illustrated in Table 2.

Masked Language Model scores
(features: src_ppl, hyp_ppl, diff_ppl)

According to Lau et al. (2017), language models (LMs) can be effectively used to estimate linguistic acceptability judgements. Salazar et al. (2020) showed that pseudo-log-likelihood scores (PLLs) and their corresponding pseudo-perplexities (PPPLs) derived from masked language models (MLMs) can help to distinguish linguistically acceptable from unacceptable sentences in an unsupervised way with comparable performance to large unidirectional autoregressive LMs. Based on the above observation, our primary objective is to derive scores at the sentence level that reflect the overall likelihood that the model gives to an entire sentence. We choose the multilingual XLM-RoBERTa[1] model and compute PLL scores by iteratively masking all tokens of the sequence. We generate PLL scores for both the source and the hypothesis and then we also calculate their absolute difference.

NMT Model scores
(features: model_scores)

According to Fomicheva et al. (2020c), seq2seq NMT models can provide meaningful insights for measuring the model's uncertainty that can be effectively used to estimate translation quality. At each timestep, the NMT system returns the probability distribution for every token in the sequence by applying a softmax function over the target language vocabulary. The token-level probabilities are then used to compute a sentence-level log-likelihood score. In our implementation we extracted this feature directly from the MLQE-PE dataset (Fomicheva et al., 2020a,c). Even if this information is already included in the provided dataset, we also decided to build another model, similar to the one described in this paper, that predicts these specific values when there is no access to the NMT system used to produce the translations.

Independent NMT Model scores
(features: M2M100_loss)

We use the pretrained M2M100 multilingual seq-to-seq model (Fan et al., 2020) to re-score the provided NMT outputs for each sentence pair. Our objective is not to generate a new hypothesis for each source sentence, but to compare every given hypothesis with the prediction produced by another multilingual translation system. The final score corresponds to the calculated cross-entropy loss when comparing the generated prediction of the M2M100 model to the provided NMT hypothesis.

Semantic Textual Similarity scores
(features: cos_sim)
Sentence similarity corresponds to the task of automatically identifying how similar or dissimilar two texts are. Neural models compare sentences by initially transforming them into semantic vectors, also known as sentence embeddings. We use the LaBSE[4] (Feng et al., 2022) pretrained model through the sentence transformers library (Reimers and Gurevych, 2019, 2020) to obtain a vector representation for every source sentence and its hypothesis. Then we compare their embeddings and get a cosine similarity score at sentence level.

COMET scores
(features: COMET_qe)
COMET (Rei et al., 2020) is a multilingual MT quality evaluation framework that demonstrates high correlation with human judgements. In our implementation, we use the reference-free wmt21-comet-qe-mqm[5] model (Rei et al., 2021), pretrained based on the MQM benchmark, which can be computed automatically without having available any reference translation. In this way, we are able to get one predicted score for every sentence and use this value as an additional feature during the training of our model.

HTER scores
(features: hter_scores)
The translation edit rate (TER) (Snover et al., 2006) calculates the editing operations needed to transform an MT output into a version that exactly matches at least one candidate translation among a list of gold-standard reference texts. The human-targeted translation edit rate (HTER) (Snover et al., 2006) is another version of the TER metric which incorporates the human factor in the process and requires human post-edits of the MT output. Even if this information is already included in the MLQE-PE dataset, we use the available HTER annotations in order to train another model, similar to the one described in this paper, that estimates the post-editing effort by predicting HTER scores for each source sentence and its translation. We finally use this information as an additional external feature for our final model.

The *Spearman* and *Pearson* correlation between all the aforementioned features and the target variable can be found in the Table 2. Based on these values, it seems that the features with the highest correlation are the cosine similarity (cos_sim), the COMET qe (COMET_qe), and the human-targeted translation edit rate (hter_scores). The NMT model scores (model_scores) and the independent NMT model scores (M2M100_loss) also demonstrate a moderate correlation with the z-standardized DA scores, while the masked language model scores (src_ppl, hyp_ppl, diff_ppl) have quite lower correlation comparing to the rest.

features	***Spearman* r**	***Pearson* r**
src_ppl	-0.15	-0.16
hyp_ppl	-0.14	-0.14
diff_ppl	-0.11	-0.13
M2M100_loss	-0.29	-0.25
cos_sim	0.34	0.40
COMET_qe	0.42	0.41
model_scores	0.25	0.30
hter_scores	-0.37	-0.37

Table 2: *Spearman* and *Pearson* correlation between the external selected features and the z-standardized DA scores. Features are described one by one in section 3.2.

3.3 Monte Carlo dropout

Dropout refers to randomly dropping nodes while training a neural network (Srivastava et al., 2014) and it is an effective strategy to prevent a model from overfitting. During training we use Monte Carlo dropout with a rate of 0.1 to mask random neurons of the model. Likewise, during inference we perform numerous iterations for each test instance and in this way we obtain a different score each time for the same instance by applying Monte Carlo dropout. Then, we use all the model's estimates to get an average score for every single sentence.

4 Experimental Results

In this section we present the performance of our model on the provided test dataset for the WMT 2022 shared task on quality evaluation for the prediction of sentence-level direct assessments. In particular, our model outperformed the baseline system in terms of *Spearman* and *Pearson* correlation in all the multilingual and bilingual tasks, in which we participated, as illustrated in the Tables 3 and 4 respectively. In the multilingual (full) sub-

[4] `https://huggingface.co/sentence-transformers/LaBSE`

[5] `https://github.com/Unbabel/COMET/blob/master/METRICS.md`

Model	Multi (full)	Multi (w/o en–yo)	en–cs	en–ja	en–mr	km–en
our model	0.448	0.506	0.563	0.276	0.444	0.623
baseline model	0.415	0.497	0.560	0.272	0.436	0.579

Table 3: *Spearman*'s correlations of the 2022 sentence-level DA subtask

Model	Multi (full)	Multi (w/o en–yo)	en–cs	en–ja	en–mr	km–en
our model	0.455	0.535	0.592	0.281	0.586	0.618
baseline model	0.393	0.511	0.576	0.273	0.525	0.568

Table 4: *Pearson*'s correlations of the 2022 sentence-level DA subtask

task we were ranked 3rd while in the multilingual (w/o en–yo) we got the 4th place.

Based on the official results, it seems that the lowest performing language pair, for both our model and the baseline, is English–Japanese while the highest performing one is Khmer–English. We did not further examine the reasons of this pattern, as we considered this exercise out of the scope of our study. In a future work, it would be useful to investigate whether certain factors contribute to this pattern (such as the source and target language complexity, the writing script, the performance of the pretrained models used to generate the features for each language, or even the content of the test dataset).

It is also worth mentioning that for most language pairs of the test sets, as illustrated in the Table 1, we did not have available training data. If our model had explicitly seen all of these languages during training, we would expect its performance to be improved.

The results of the test set from the official leaderboard[6] for each language pair, in which we participated, can be found in the Official results of the WMT 2022 QE Task 1 – Sentence-level Direct Assessment. In these tables, our proposed model is compared to the baseline system in terms of RMSE, MAE, *Spearman* and *Pearson* correlation coefficient.

5 Conclusions

This paper presents our submission to the WMT 2022 quality estimation Task 1 on sentence-level direct assessment. We introduce a model trained based on the predictor–estimator architecture using the XLM-RoBERTa[1] for feature prediction and a regression head to finally estimate the z-standardized DA values. We suggest the use of additional external features that reflect different criteria of human judgements and multiple levels of translation quality. These features exhibit good correlation with the target variable and consequently with human annotations. Our approach is applicable in multilingual settings even with languages or writing scripts not explicitly seen during the training of the MTQE model. Our system demonstrates competitive results and a strong correlation with human judgements of quality assessment outperforming the baseline system in terms of both *Spearman* and *Pearson* correlation coefficient.

6 Acknowledgements

The work reported in this paper was performed as part of the final master's thesis titled "Quality Estimation for Neural Machine Translation" for the completion of the MSc in Language Technology, a degree co-organized by the National Kapodistrian University of Athens and the Institute for Language and Speech Processing of the Athens Research Center. We would like to thank Welocalize and especially David Clarke for greatly supporting the development of this MSc thesis.

[6] https://www.statmt.org/wmt22/quality-estimation-task_results.html

References

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence Estimation for Machine Translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 315–321, Geneva, Switzerland. COLING.

Yimeng Chen, Chang Su, Yingtao Zhang, Yuxia Wang, Xiang Geng, Hao Yang, Shimin Tao, Guo Jiaxin, Wang Minghan, Min Zhang, Yujia Liu, and Shujian Huang. 2021. HW-TSC's Participation at WMT 2021 Quality Estimation Shared Task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 890–896, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond English-Centric Multilingual Machine Translation.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT Sentence Embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Erick Fonseca, Frédéric Blain, Vishrav Chaudhary, Francisco Guzmán, Nina Lopatina, Lucia Specia, and André F. T. Martins. 2020a. MLQE-PE: A Multilingual Quality Estimation and Post-Editing Dataset. *arXiv preprint arXiv:2010.04480*.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Vishrav Chaudhary, Mark Fishel, Francisco Guzmán, and Lucia Specia. 2020b. BERGAMOT-LATTE Submissions for the WMT20 Quality Estimation Shared Task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1010–1017, Online. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020c. Unsupervised Quality Estimation for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES Evaluation Datasets for Low-Resource Machine Translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Hyun Kim and Jong-Hyeok Lee. 2016. A Recurrent Neural Networks Approach for Estimating the Quality of Machine Translation Output. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 494–498, San Diego, California. Association for Computational Linguistics.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-Estimator using Multilevel Task Learning with Stack Propagation for Neural Quality Estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 562–568, Copenhagen, Denmark. Association for Computational Linguistics.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, Acceptability, and Probability: A Probabilistic View of Linguistic Knowledge. *Cognitive science*, pages 1202–1241.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*. arXiv.

João Moura, Miguel Vera, Daan van Stigt, Fabio Kepler, and André F. T. Martins. 2020. IST-Unbabel Participation in the WMT20 Quality Estimation Shared Task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1029–1036, Online. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. TransQuest: Translation Quality Estimation with Cross-lingual Transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5070–5081, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ricardo Rei, Ana C Farinha, Chrysoula Zerva, Daan van Stigt, Craig Stewart, Pedro Ramos, Taisiya Glushkova, André F. T. Martins, and Alon Lavie. 2021. Are References Really Needed? Unbabel-IST 2021 Submission for the Metrics Shared Task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1030–1040, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A Neural Framework for MT Evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked Language Model Scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2018. RUSE: Regressor Using Sentence Embeddings for Automatic Machine Translation Evaluation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 751–758, Belgium, Brussels. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Lucia Specia, Frédéric Blain, Varvara Logacheva, Ramón F. Astudillo, and André F. T. Martins. 2018. Findings of the WMT 2018 Shared Task on Quality Estimation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 689–709, Belgium, Brussels. Association for Computational Linguistics.

Lucia Specia and Kashif Shah. 2018. *Machine Translation Quality Estimation: Applications and Future Perspectives*, pages 201–235. Springer International Publishing, Cham.

Lucia Specia, Marco Turchi, Nicola Cancedda, Nello Cristianini, and Marc Dymetman. 2009. Estimating the Sentence-Level Quality of Machine Translation Systems. In *Proceedings of the 13th Annual conference of the European Association for Machine Translation*, pages 28–35, Barcelona, Spain. European Association for Machine Translation.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Jiayi Wang, Ke Wang, Boxing Chen, Yu Zhao, Weihua Luo, and Yuqi Zhang. 2021. QEMind: Alibaba's Submission to the WMT21 Quality Estimation Shared Task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 948–954, Online. Association for Computational Linguistics.

Chrysoula Zerva, Daan van Stigt, Ricardo Rei, Ana C Farinha, Pedro Ramos, José G. C. de Souza, Taisiya Glushkova, Miguel Vera, Fabio Kepler, and André F. T. Martins. 2021. IST-Unbabel 2021 Submission for the Quality Estimation Shared Task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 961–972, Online. Association for Computational Linguistics.

7 Official results of the WMT 2022 QE Task 1 – Sentence-level Direct Assessment

Model	***Spearman*** r	***Pearson*** r	**RMSE**	**MAE**	**Disk footprint (bytes)**	**Model params.**
baseline	0.415	0.393	0.979	0.820	2,280,011,066	564,527,011
our model	0.448	0.455	0.794	0.632	2,307,101,417	576,733,248

Table 5: Evaluation of the **Multilingual models** in the 2022 DA subtask

Model	***Spearman*** r	***Pearson*** r	**RMSE**	**MAE**	**Disk footprint (bytes)**	**Model params.**
baseline	0.497	0.511	0.748	0.585	2,280,011,066	564,527,011
our model	0.506	0.535	0.733	0.571	2,307,068,585	576,725,041

Table 6: Evaluation of the **Multilingual models (without en–yo)** in the 2022 DA subtask

Model	***Spearman*** r	***Pearson*** r	**RMSE**	**MAE**	**Disk footprint (bytes)**	**Model params.**
baseline	0.560	0.576	0.804	0.608	2,280,011,066	564,527,011
our model	0.563	0.592	0.785	0.610	2,307,068,585	576,725,041

Table 7: Evaluation of the **en–cs models** in the 2022 DA subtask

Model	***Spearman*** r	***Pearson*** r	**RMSE**	**MAE**	**Disk footprint (bytes)**	**Model params.**
baseline	0.272	0.273	0.747	0.576	2,280,011,066	564,527,011
our model	0.276	0.281	0.755	0.579	2,307,068,585	576,725,041

Table 8: Evaluation of the **en–ja models** in the 2022 DA subtask

Model	***Spearman*** r	***Pearson*** r	**RMSE**	**MAE**	**Disk footprint (bytes)**	**Model params.**
baseline	0.436	0.525	0.628	0.461	2,280,011,066	564,527,011
our model	0.444	0.586	0.534	0.401	2,307,068,585	576,725,041

Table 9: Evaluation of the **en–mr models** in the 2022 DA subtask

Model	***Spearman*** r	***Pearson*** r	**RMSE**	**MAE**	**Disk footprint (bytes)**	**Model params.**
baseline	0.579	0.568	0.774	0.616	2,280,011,066	564,527,011
our model	0.623	0.618	0.794	0.619	2,307,068,585	576,725,041

Table 10: Evaluation of the **km–en models** in the 2022 DA subtask

Edinburgh's submission to the WMT 2022 Efficiency task

Nikolay Bogoychev[†] **Biao Zhang**[†], **Maximiliana Behnke**[†] **Graeme Nail**[†]
Jelmer van der Linde[†] **Sidharth Kashyap**[‡] **Kenneth Heafield**[†]
[†]University of Edinburgh [‡]Intel Corporation
{n.bogoych, biao.zhang, maximiliana.behnke, graeme.nail, jelmer.vanderlinde
kenneth.heafield}@ed.ac.uk, sidharth.n.kashyap@intel.com

Abstract

We participated in all tracks of the WMT 2022 efficient machine translation task: single-core CPU, multi-core CPU, and GPU hardware with throughput and latency conditions. Our submissions explore a number of efficiency strategies: knowledge distillation, a simpler simple recurrent unit (SSRU) decoder with one or two layers, shortlisting, deep encoder, shallow decoder, pruning and bidirectional decoder. For the CPU track, we used quantized 8-bit models. For the GPU track, we used FP16 quantisation. We explored various pruning strategies and combination of one or more of the above methods.

1 Introduction

This paper describes the University of Edinburgh's submission to Seventh Conference on Machine Translation (WMT2022) Efficiency Task[1], which measures performance on latency and throughput on both CPU and GPU, in addition to translation quality. Our submission focused on the trade-off between these metrics and quality.

Our submission builds upon the work of last year's submission (Behnke et al., 2021). We trained our models in a teacher-student setting (Kim and Rush, 2016), using the data provided by the organisers. For the students, we used a Simpler Simple Recurrent Unit (SSRU) (Kim et al., 2019) decoder, used a target vocabulary shortlist, and experimented with pruning the student models by removing component and block-level parameters to improve speed. We used 8-bit quantisation for the CPU submission and FP16 quantisation for the GPU submission. We further experimented with IBDecoder (Zhang et al., 2020).

For running our experiments, we improved upon the Marian (Junczys-Dowmunt et al., 2018) machine translation framework by incorporating speed ups for 8-bit matrix multiplication operations, optimizations for pruning neural network parameters on Intel CPUs, and profiler aided optimisation of various components.

1.1 Efficiency Shared Task

The WMT22 efficiency shared task consists of two sub-tasks: throughput and latency. Systems should translate English to German under the constrained conditions, where the teacher model and the distilled data are provided. For each task, systems are provided 1 million lines of raw English input with at most 150 space-separated words. The throughput task receives this input directly. The latency task, introduced in WMT21, is fed input one sentence at a time, waiting for the translation output before providing the next sentence.

Throughput is measured on multi-core CPU or GPU system, and latency is measured on single-core CPU or GPU systems. The CPU-based evaluations use an Intel Ice Lake system via Oracle Cloud BM.Optimized3.36, while the GPU-based use a single A100 via Oracle Cloud BM.GPU4.8.

Entries to both tasks are measured on quality, approximated via COMET score (Rei et al., 2020), speed, model size, Docker image size, and memory consumption. We did not optimise specifically for the latency task beyond configuring the relevant batch sizes to one. We used Ubuntu 22.04 based images for our systems, with standard Ubuntu for CPU-only systems and NVIDIA's Ubuntu-based CUDA-11.7 docker for GPU-capable systems. Docker images were created using multi-stage builds, with model disk size reduced by compression with xzip.

2 Knowledge distillation

We used the provided distilled data to build different student systems. The provided data was distilled through two different processes; for the monolingual input, distilled data was generated using a

[1] http://statmt.org/wmt22/efficiency-task.html

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 661 - 667
December 7-8, 2022 ©2022 Association for Computational Linguistics

beam-size of 6. For the parallel data, the teacher ensemble was used to produce the 6 best candidate translations for each input sentence, the candidate most similar to the parallel reference in BLEU score was kept as the distilled sentence. We trained student models using just the provided parallel data and identical systems using parallel+monolingual data. Our early comparisons showed that the full corpora produced higher quality student systems according to automatic metrics; submitted systems therefore use both parallel and monolingual data.

The student models were trained using a validation set consisting of the subset of sentences in the English-German WMT test sets from 2014–2019 that were originally in English. Training concluded after reaching 20 consecutive validations without an improvement in BLEU score. The student models all used the provided shared SentencePiece vocabulary. We used the default training hyperparameters from Marian for the transformer-base model with the learning rate reduced to 0.0002.

We explored a number of different configuration in order to find the optimal system on the Pareto frontier for speed-quality. We experimented with the following configurations:

Deep encoders/Shallow decoders The majority of the computational cost of the machine translation system falls to the decoder. We can therefore increase drastically the number of encoder layers and decrease the decoder layers without noticeable drop in quality (Kong et al., 2021).

Tied decoder layers Since matrix multiplication is a memory bound problem, we can increase the number of decoder layers, as long as we don't add extra parameters. Tied decoder layers allow us to maintain the same memory footprint and keep all the traversed matrices in cache.

SSRU We replace the self-attention in the decoder with an RNN using the less computation and memory intensive cell SSRU.

Reduced model dimensions We reduce the model dimensions, using several presets. See table 1 for details.

Wide embeddings We increased the size of the embedding dimension to match the FFN dimension. While this produces models that are strictly larger than their non-wide equivalent, the initial increased capacity can yield competitive systems using fewer layers.

Fewer heads We reduced the number of attention heads for some of the smaller models. These have the same number of parameters, but intermediate computations have different shape inputs.

3 Pruning

Attention is a crucial part of the transformer architecture, but it is also computationally expensive. Research has shown that many heads can be pruned after training; with further work suggesting that pruning during training can be less damaging to quality. Feedforward layers are also expensive and could be reduced.

We expand upon our work from the previous year on the group lasso regularisation. We build upon the standard group lasso with a novel approach of *aided regularisation*. The idea behind it is to use supplementary information to scale the penalties per layer to steer them towards a specific behaviour. In practice, it means adding a new scalar γ alongside an already existing λ:

$$E(batch) = \frac{1}{|batch|}\left(\sum_{x \in batch} CE(x) + \lambda \sum_{l \in layers} \gamma_l^{batch} R(l)\right)$$

As shown in the equation above, each layer has its individual γ, which gets updated after every backpropagation pass. In order to avoid sudden shits in γ between individual batches, which could make a ratio between perplexities and penalties even more unstable, γ are exponentially smoothed as training progresses:

$$\gamma^j \leftarrow \alpha\gamma^j + (1 - \alpha) * \gamma^{j-1}$$

After every batch i, we calculate a local scalar γ_j for each layer j based on information gathered during this specific update, which then updates a smoothed global scalar. α is a constant used in exponential average that controls the contribution of a new element in a sequence towards the overall average. We use $\alpha = 1\mathrm{e}{-4}$ in my experiments.

We explore *gradient-aided regularisation* which *scales penalties based on layer gradients*. γ scalars should increase as gradients stop flowing through a layer since it indicates that this layer does not contribute to training as much, possibly stopping learning altogether. A layer with small gradients is a good candidate to be regularised more aggressively and vice versa.

With W_i being a regularised layer and ∇W as accumulated gradients in a model, the gradient-aided γ function is defined as:

	Layers		Dimensions			Size		Quality			Speed
Model	Encoder	Decoder	Emb.	FFN	Att. heads	Params	Disk	BLEU	chrF	COMET	Time
Teacher	6	6	1024	4096	16	627.5M	3.19GB	43.37	67.39	0.5908	—
Large	12	1	1024	3072	8	171.4M	654MB	44.26	68.06	0.5901	170.4
Base	12	1	512	2048	8	57.9M	222MB	44.06	67.94	0.5842	57.7
Tiny	12	1	256	1536	8	22.0M	85MB	43.32	67.36	0.5516	23.4
Micro	12	1	256	1024	8	18.6M	72MB	43.00	67.16	0.5389	20.9
Base	6	2	512	2048	8	42.7M	163MB	44.04	67.90	0.5879	50.5
Tiny	6	2	256	1536	8	16.9M	65MB	42.76	67.12	0.5538	19.6
Tied.Tiny	6	2	256	1536	8	15.7M	61MB	42.72	67.08	0.5470	17.7
Tied.Tiny	8	4	256	1536	8	17.8M	69MB	43.22	67.38	0.5621	23.0
Base.Wide	12	1	2048	2048	8	401.5M	1.50GB	43.82	67.74	0.5773	395.4
Base.Wide	6	2	2048	2048	8	283.8M	1.1GB	44.28	68.14	0.5979	374.7

Table 1: Architectures for the different student models. The number of encoder/decoder layers are reported with the size of the embedding and FFN layers, the total number of parameters and the model size on disk. Quality and speed evaluated and averaged across WMT16–19.

	Layers		Sparsity		Quality			Speed
Model	Encoder	Decoder	Attention	FFN	BLEU	chrF	COMET	Time
Base	12	1	0%	0%	44.06	67.94	0.5842	57.7
+ pruning	12	1	63%	20%	43.92	67.86	0.5825	44.6
+ pruning + ft8bit	12	1	63%	20%	43.68	67.66	0.5710	18.6
Tiny	12	1	0%	0%	43.32	67.36	0.5516	23.4
+ pruning	12	1	74%	72%	41.54	66.16	0.4882	12.3
+ pruning + ft8bit	12	1	74%	72%	41.02	65.70	0.4615	5.8
Tied Tiny	8	4	0%	0%	43.22	67.38	0.5621	23.0
+ pruning	8	4	46%	20%	42.98	67.22	0.5584	19.3
+ pruning + ft8bit	8	4	46%	20%	42.36	66.78	0.5393	10.0

Table 2: The evaluation of student models pruned with aided regularisation and quantised to 8-bits. Both quality and speed has been averaged over WMT16–20 testsets. Quantised models were finetuned shortly to help recover quality.

$$\gamma_i = -log\left(\frac{\|\frac{\partial W_i}{\partial E}\|_2}{\|\nabla W\|_2}\right)$$

We follow the training regime outlined by Behnke et al. (2021):

1. Pretraining (50k batches)
2. Regularise (200/300k batches)
3. Slice and converge (200k+ batches)

All on-going training statistics including the learning rate and Adam optimiser were refreshed after each step. The results are presented in Tab. 2. The results include quantised inference with models finetuned for the best quality performance. The 12-1.Base model was regularised for 300k batches with $\lambda = 0.05$. The 12-1.Tiny model was regularised for 200k batches with $\lambda = 0.5$. Both aforementioned models were pruned in the encoder only. The 8-4.Tiny.Tied model was regularised for 200k batches with $\lambda = 0.3$ with both encoder and decoder layers being penalised.

The quality gap becomes larger the harsher pruning is. The base transformer model with 12 pruned encoder layers gets 1.3× faster at the cost of 0.0017 COMET point. Applying quantisation on the top of it makes translation 3.1× faster in exchange of 0.13 COMET points.

We applied regularisation onto both encoder and decoder with the "8-4" tied tiny transformer architecture. This pruned and quantised model speeds up by a factor of 2.3× at a 0.8 BLEU drop.

The most aggressive pruning among the presented results is a tiny transformer with 12 encoder layers with more than 70% parameters removed. This model is 4× faster in comparison to its baseline with 3.3 BLEU and 0.09 COMET points drop.

We note that quantisation struggles with quality on smaller models, both when trained from scratch or pruned. Fine-tuning rectifies the problem to some degree, but quality is sacrificed for faster translation in the end.

4 Fixed Point 8-bit Quantisation

Quantising FP32 models into 8-bit integers is a known strategy to reduce decoding time, specifically on CPU, with a minimal impact on quality (Kim et al., 2019; Bhandare et al., 2019; Rodriguez et al., 2018). This year's submission closely follows the quantisation scheme of last year's work (Behnke and Heafield, 2021).

Quantisation entails computing a scaling factor to collapse the range of values to $[-127, 127]$. For parameters, this scaling factor is computed offline using the maximum absolute value but activation tensors change at runtime. To compute a scaling factor for them, we decoded the WMT16-20 datasets and recorded the scaling factor $\alpha(A_i) = 127/max(|A_i|)$ for each instance A_i of an activation tensor A. Then, for production, we fixed the scaling factor for activation tensor A to the mean scaling factor plus 1.1 standard deviation: $\alpha(A) = \mu(\{\alpha(A_i)\}) + 1.1 * \sigma(\{\alpha(A_i)\})$. These scaling factors were baked into the model file so that statistics were not computed at runtime.

We used predominantly *intgemm*[2] for our 8-bit GEMM operations, including for the shortlisted output layer. All parameter matrices are quantised to 8-bit offline and the activations get quantised dynamically before a GEMM operation. We only perform the GEMM operation and the following activation in 8-bit integer mode. After a GEMM operation, the output is de-quantized back to FP32. More formally we perform $dequantize(\sigma(A{*}B{+}bias))$, where the addition of the $bias$, the activation function σ, and the de-quantisation are applied in a streaming fashion to prevent a round trip to memory.

Furthermore we make use of Intel's *DNNL*[3] for our pruned models, as it performs better than *intgemm* for irregular sized matrices. Unfortunately, *DNNL* doesn't support streaming de-quantisation, bias addition or activation function application.

Quantisation does not extend to the attention layer, which is still computed in FP32. The reason being is that in the attention layer, both the A and B matrices of the GEMM operation would need to be quantised at runtime, which makes the quantisation too expensive. We note that we only perform the GEMM operations in 8-bit integers.

Similar to previous' year's submission, we performed quantisation fine-tuning for some 8-bit models, where we perform a small amount of training with low learning rate and a damaged GEMM implementation that simulates the quantised output. We found that this helps regain some quality, especially in smaller models.

5 Shortlisting

The single most expensive computation in machine translation is the cost of the output layer. We can reduce the computation if we only take into account likely output tokens, reducing the output layer size from 32000 to something much more manageable like 500-2000. We used IBM model based shortlisting (Kim et al., 2019).

This lexical shortlist is straightforward to work with, but it is limiting in the sense that it doesn't capture well idioms and favours more literal translations. The hyper-parameters that control the size of this shortlist are: the number of most frequently targeted words included, and the number of probable translations for each token in the input. This year we increased the number of most-frequent and aligned tokens to `100,100` (from `50,50` in the previous year) in order to improve quality.

The shortlist is built using alignment models trained on a specific corpora. The total number of tokens in a shortlist considered is influenced by the size of the current batch: The shortlist produced is the union of probable translations for each input token and overall most-likely candidates. In latency scenarios, where batches are a single sentence, a small shortlist is more detrimental to the quality than for larger batches, such as in throughput scenarios, that benefit from inclusion of more candidate tokens. Similarly, this approach benefits when inputs are batched.

6 IBDecoder

The sequential nature of autoregressive decoding forms an inference bottleneck, hurting decoding parallelisation and latency. A popular method to break this bottleneck is to allow the parallel prediction of multiple target tokens per step through semi- or non-autoregressive modelling (Gu et al., 2018; Wang et al., 2018) with a quality tradeoff.

We experimented with Interleaved Bidirectional Decoder (Zhang et al., 2020), a variant of semi-autoregressive decoder that predicts target tokens from the left-to-right and the right-to-left directions

[2] `https://github.com/kpu/intgemm`

[3] `https://github.com/oneapi-src/oneDNN`

	Layers		Size		Quality			Speed
Model	Encoder	Decoder	Params	Disk	BLEU	chrF	COMET	Time
Base	12	1	57.9M	222MB	44.06	67.94	0.5842	57.7
+ IBDecoder	12	1	57.9M	221MB	43.84	67.74	0.5605	51.6
+ 8bit quantisation	12	1	57.9M	221MB	43.50	67.48	0.5412	28.6
Base + IBDecoder	12	1	57.9M	221MB	43.84	67.74	0.5605	51.6
+ pruning	12	1	57.9M	168MB	43.50	67.48	0.5340	42.1
+ 8bit quantisation	12	1	57.9M	168MB	43.26	67.28	0.5166	18.8
Tiny	6	2	16.9M	65MB	42.76	67.12	0.5538	19.6
+ IBDecoder	6	2	16.9M	65MB	41.88	66.64	0.5074	17.1
+ 8bit quantisation	6	2	16.9M	65MB	41.00	65.94	0.4628	9.8
Tiny + IBDecoder	6	3	18.1M	69MB	42.48	66.98	0.5275	19.6
+ 8bit quantisation	6	3	18.1M	69MB	41.62	66.32	0.4971	11.2
Micro + IBDecoder	12	4	21.4M	82MB	42.96	67.28	0.5475	25.3
+ 8bit quantisation	12	4	21.4M	82MB	42.56	67.08	0.5338	15.4

Table 3: The evaluation of IBDecoder models and their 8-bit quantisation (without finetuning). Both quality and speed has been averaged over WMT16–20 testsets.

simultaneously. Zhang et al. (2020) showed that words from different directions are more loosely dependent thus their parallel generation hurts quality less. IBDecoder produces one word in each direction at a time, thus halving the total decoding steps and approximately doubling speed.

The efficiency gains from IBDecoder decrease when using deep encoders and shallow decoders (Tab. 3). In general, IBDecoder delivers a speed-up over our baseline system and is competitive at BLEU scores but much worse at COMET scores. IBDecoder shows higher sensitivity to model sizes, where reducing model size dramatically hurts its performance regardless of BLEU or COMET. We also tried to initialise IBDecoder from the baseline system which unfortunately doesn't help. IBDecoder also benefits from pruning and quantisation in speed, but at the cost of losing COMET.

7 Quality issues

Quantisation applied to small models, especially those that were pruned, struggles with maintaining the quality. For example, as can be seen in Tab. 2, quantisation on top of pruned models damages the quality from 0.1 to 0.3 COMET points. This gap is more evident in smaller architectures such as Tiny or Tied Tiny. We hypothesise that the fewer parameters there are in a model, the more difficult it is to optimise through pruning and/or quantisation, or using a bidirectional generation.

IBDecoder suffers from the pruning and quantisation particularly on the COMET scores as shown in Tab. 3.

We compared several sentences that showed little difference in BLEU but significant difference in the COMET scores and tried to see what went wrong (Table 4). We can see that the IBDecoder is prone to pathological repetitions (Example 1) and even more so when quantised (Examples 3 and 4). Those repetitions, especially long ranged one don't hurt the BLEU score, but they get heavily penalised by the COMET score (Example 3).

It seems quantisation doesn't always result in a a worse transaltion. In the second example the quantised IBDecoder produces a more complicated, but overall much better translation than the IBDecoder, which also suffers from a repetition error. This suggests that the model is quite brittle and very susceptible to small changes.

8 Software improvements

We built our work using the Marian machine translation framework, making some improvements on top of the submission from last year:

AVX512 inrinsics We implemented hand crafted intrinsics for various arithmetic operations, resulting in .5% improvement in performance.

Max element We identified via a profiler that the *max element* implementation was taking more time than usual so we implemented a hand optimised version resulting in 5% performance improvement. More details are available in Appendix A.

Thread configuration For the CPU_ALL throughput track, we swept configurations of multiple processes and threads on the platform, settling

Reference	Biotechnische Anwendungen
Baseline	Biotech-Anwendungen (0.6632)
IBDecoder	Anwendungen in der-Anwendungen (-1.4205)
IBDecoder-Quant	Anwendungen in der-Anwendungen (-1.4205)
Reference	Die nächste Show findet am 9. Oktober in San Francisco statt. Am 16. März 2020 wird die Band ihre UK-Tournee in Manchester eröffnen.
Baseline	Ihre nächste Show ist am 9. Oktober in San Francisco und die Band wird ihre UK-Tour in Manchester am 16. März 2020 eröffnen. (0.7350)
IBDecoder	Ihr nächster Auftritt ist am 9. Oktober in San Francisco und eröffnet eröffnet ihre UK-Tour in Manchester am 16. März 2020. (-0.1582)
IBDecoder-Quant	Ihre nächste Show findet am 9. Oktober in San Francisco statt, wo die Band ihre UK-Tournee in Manchester am 16. März 2020 eröffnen wird. (0.7387)
Reference	Die Herzogin von York schrieb auf Twitter: „ Ich kenne die Gefühle einer Mutter, deshalb weine ich vor Freude. Ich freue mich sehr über diese sensationellen Neuigkeiten
Baseline	Die Herzogin von York schrieb auf Twitter: "Ich weiß, was eine Mutter fühlt, also habe ich Tränen der Freude. (0.4349)
IBDecoder	Die Herzogin von York schrieb auf Twitter: "Ich weiß, was eine Mutter fühlt, also habe ich Tränen der Freude. (0.4351)
IBDecoder-Quant	Die Herzogin von York schrieb auf Twitter: "Ich weiß, was eine Mutter Freude, also habe ich Tränen der Freude, also habe ich Tränen der Freude. (-0.9820)
Reference	Meghan Markle bezüglich des Kurzauftritts bei Suits „nie gefragt“
Baseline	Meghan Markle wurde nach Suits Cameo "nie gefragt" (0.1344)
IBDecoder	Meghan Markle wurde "niemals" nach Suits Cameo gefragt (0.0509)
IBDecoder-Quant	Meghan Markle wurde "nicht gefragt" nach Suits Cameo gefragt (-1.1194)

Table 4: Case study for IBDecoder models. All models are with 12 encoder layers and 1 decoder layer under the base setup. IBDecoder-Quant denotes the quantised system. We show cases where IBDecoder and IBDecoder-Quant performs worse than Baseline and IBDecoder, respectively, and the numbers in bracket shows the COMET scores.

on 4 processes with 9 threads each. The input text is simply split into 4 pieces and parallelised (Tange, 2011) over processes. The mini-batch sizes past 16 did not impact performance substantially but 32 was chosen as the best performing one. The Hyperthreads do not increase performance. Each process is bound to 9 cores assigned sequentially and to the memory domain corresponding to the socket with those cores using *numactl*. Output from the data parallel run is then stitched together to produce the final output.

For our GPU submission we reused the work from the last year's submission (Behnke et al., 2021) with the improved models.

9 Conclusion

We participated in all tracks of the WMT 2022 efficiency task and we submitted multiple systems that have different trade-offs between speed and translation quality. For the CPU submission we used 8-bit integer decoding and a combination of pruned and non-pruned system, together with a lexical shortlist in order to reduce the computational cost of the output layer. We also experimented with IBDecoder in both CPU and GPU setting.

Acknowledgements

The computations described in this research were performed using the Baskerville Tier 2 HPC service (https://www.baskerville.ac.uk/). Baskerville was funded by the EPSRC and UKRI through the World Class Labs scheme (EP/T022221/1) and the Digital Research Infrastructure programme (EP/W032244/1) and is operated by Advanced Research Computing at the University of Birmingham.

References

Maximiliana Behnke, Nikolay Bogoychev, Alham Fikri Aji, Kenneth Heafield, Graeme Nail, Qianqian Zhu, Svetlana Tchistiakova, Jelmer van der Linde, Pinzhen Chen, Sidharth Kashyap, and Roman Grundkiewicz. 2021. Efficient machine translation with model pruning and quantization. In *Proceedings of the Sixth Conference on Machine Translation*, pages 775–780, Online. Association for Computational Linguistics.

Maximiliana Behnke and Kenneth Heafield. 2021. Pruning neural machine translation for speed using group lasso. In *Proceedings of the Six Conference on Machine Translation*, Online. Association for Computational Linguistics.

Aishwarya Bhandare, Vamsi Sripathi, Deepthi Karkada, Vivek Menon, Sun Choi, Kushal Datta, and Vikram

Saletore. 2019. Efficient 8-bit quantization of transformer neural machine language translation model.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, et al. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Young Jin Kim, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019. From research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 280–288, Hong Kong. Association for Computational Linguistics.

Xiang Kong, Adithya Renduchintala, James Cross, Yuqing Tang, Jiatao Gu, and Xian Li. 2021. Multilingual neural machine translation with deep encoder and multiple shallow decoders. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1613–1624, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Andres Rodriguez, Eden Segal, Etay Meiri, Evarist Fomenko, Young Jin Kim, Haihao Shen, and Barukh Ziv. 2018. Lower numerical precision deep learning inference and training.

O. Tange. 2011. Gnu parallel - the command-line power tool. *;login: The USENIX Magazine*, 36(1):42–47.

Chunqi Wang, Ji Zhang, and Haiqing Chen. 2018. Semi-autoregressive neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 479–488, Brussels, Belgium. Association for Computational Linguistics.

Biao Zhang, Ivan Titov, and Rico Sennrich. 2020. Fast interleaved bidirectional sequence generation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 503–515, Online. Association for Computational Linguistics.

A Profiler aided optimisation

We used a profiler to identify hotspots for potential software optimisations.

Max element improvements We identified that the *max element* operation takes surprisingly large amounts of runtime during decoding. *max element* is used to select the next word to be produced from the output layer during decoding with beam size of 1. We explored various different implementations and achieved 10X performance improvement compared to the standard library when using GCC and 2X otherwise. This resulted in about 5% performance improvement in the CPU setting. More details about different implementations can be seen on Table 5

	GCC	**clang**
std::max_element	2.670s	0.422s
sequential	1.083s	1.192s
AVX512 max + max_reduce	0.241s	0.215s
AVX512 max_reduce only	0.257s	0.263s
AVX512 cmp_ps_mask	0.188s	0.183s
AVX512 ^+ vectorized overhang	0.210s	0.209s
AVX cmp_ps + movemask	0.218s	0.170s
SSE cmplt_psp + movemask	0.269s	0.205s

Table 5: Performance of *max element* with GCC 11.2 and clang 14 on Intel Cascade lake. For more information check `https://github.com/XapaJIaMnu/maxelem_test`.

CUNI Non-Autoregressive System for the WMT 22 Efficient Translation Shared Task

Jindřich Helcl
Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 11800 Prague, Czech Republic
helcl@ufal.mff.cuni.cz

Abstract

We present a non-autoregressive system submission to the WMT 22 Efficient Translation Shared Task. Our system was used by Helcl et al. (2022) in an attempt to provide fair comparison between non-autoregressive and autoregressive models. This submission is an effort to establish solid baselines along with sound evaluation methodology, particularly in terms of measuring the decoding speed. The model itself is a 12-layer Transformer model trained with connectionist temporal classification on knowledge-distilled dataset by a strong autoregressive teacher model.

1 Introduction

In the past few years, non-autoregressive (NAR) models for neural machine translation (NMT) attracted interest from the research community (Gu et al., 2018; Lee et al., 2018). Given the conditional independence between the output states, the decoding process can be parallelized across time steps. In theory, this leads to higher decoding speeds.

Since efficient decoding is claimed to be the main motivation of non-autoregressive models, the Efficient Translation Shared Task seems to be the appropriate venue to provide fair comparison between these models and their autoregressive counterparts. However, all submissions to this task were autoregressive so far (Birch et al., 2018; Hayashi et al., 2019; Heafield et al., 2020, 2021).

Recently, Helcl et al. (2022) pointed out common flaws in the evaluation methodology of NAR models. We found that optimized autoregressive models still achieve superior performance over NAR models. The only scenario where NAR models showed some potential is GPU decoding with batch size of 1 (latency). Nevertheless, optimized autoregressive models were still both faster and better in terms of translation quality. The main purpose of this submission is to provide a reasonable baseline to future non-autoregressive submissions.

2 Model

In our experiments, we use the non-autoregressive model proposed by Libovický and Helcl (2018) based on Connectionist Temporal Classification (CTC; Graves et al., 2006). We submit models that have been trained as a part of Helcl (2022).

Architecture. The architecture is a 6-layer Transformer encoder, followed by a state-splitting layer and another stack of 6 Transformer layers. The state-splitting layer takes the encoder states, project them into k-times wider states using an affine transformation, and then split the states into k-times longer sequence while retaining the original model dimension. In the submitted model, we set $k = 3$. The latter 6 layers cross-attend to the states immediately after state-splitting. We use Transformer model dimension of 1,024, 16 attention heads and a dimension of 4,096 in the feed-forward sublayer.

The defining property of non-autoregressive models is that the decoding process treats output states as conditionally independent. In this architecture, we set the output sequence length to $k \times T_x$ where T_x is the length of the source sentence. To allow for shorter output sequences, the any output state can produce an empty token. The training loss is then computed using a dynamic programming algorithm as a sum of losses of all possible empty token alignments which lead to the same output sentence. The schema of the architecture is shown in Figure 1.

Training. We train our model on the knowledge-distilled data generated by the provided teacher (Chen et al., 2021). We use learning rate of 0.0001 in a inverse square-root decay scheme with 8,000 warm-up and decay steps.

Implementation. We implement and train our model in the Marian toolkit (Junczys-Dowmunt et al., 2018). For the CTC implementation,

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 668 - 670
December 7-8, 2022 ©2022 Association for Computational Linguistics

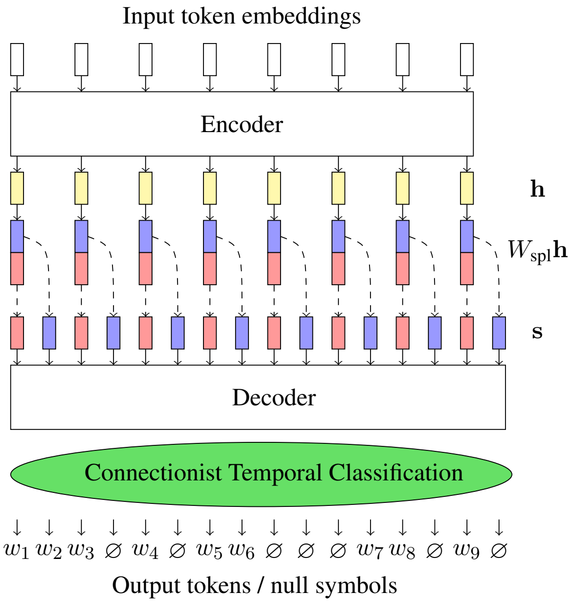

Input token embeddings
Encoder
h
$W_{\mathrm{spl}}\mathbf{h}$
s
Decoder
Connectionist Temporal Classification
w_1 w_2 w_3 ∅ w_4 ∅ w_5 w_6 ∅ ∅ ∅ w_7 w_8 ∅ w_9 ∅
Output tokens / null symbols

Figure 1: The CTC-based model architecture. We show the original image from Libovický and Helcl (2018).

we use the warp-ctc library[1]. We release our code at https://github.com/jindrahelcl/marian-dev. The trained model (and a number of different variants including models in opposite translation direction) can be downloaded at https://data.statmt.org/nar.

3 Results

We refer the reader to the original paper for more details about the evaluation and its results. The model we submitted is denoted in the paper as "large". A summary of the results follows.

Translation Quality. To summarize the main findings, the model achieves a competitive BLEU score (Papineni et al., 2002) on the WMT 14 news test set (Bojar et al., 2014), which serves as a comparison to other non-autoregressive models that use this test set as the de facto standard benchmark. When evaluated on the WMT 19 news test set, our model obtains BLEU of 47.8, and a COMET score (Rei et al., 2020) of 0.1485. Compared to an similarly-sized autoregressive teacher model with 50.5 BLEU and COMET of 0.4110, we see a somewhat surprising gap between the COMET scores while BLEU scores are relatively close. We hypothesize that the errors that the non-autoregressive model makes are out of the training domain of the COMET models, which makes them more sensitive towards this kind of errors.

Decoding Time. We evaluated our models on the one million sentences benchmark used in the previous editions of this task (Heafield et al., 2021), and we tried to reproduce the official hardware setup to large extent. For CPU decoding, we measured time to translate the test set on an Intel Xeon 6354 server from Oracle Cloud, with 36 cores. We run the evaluation only in the batch decoding mode, as the models were too slow to decode with a single sentence in batch. With the submitted model, the translation on CPU took 7,434 seconds (using batch of 16 sentences).

We used a single Nvidia A100 GPU for GPU decoding. In the latency setup, the translation took 7,020 seconds, and the batched decoding ($b = 128$) took 782 seconds. When compared with other submissions to this task, we find that the smallest difference is indeed found in the GPU decoding latency setting. However, the optimized models submitted to last year's round still achieved significantly better decoding times.

4 Conclusions

We submit a non-autoregressive system to the Efficient Translation Shared Task to the WMT 22. The model is trained with connectionist temporal classification, which allows the generation of empty tokens and thus making generation of sentences of various length possible while retaining the conditional independence among output tokens without explicit length estimation.

The main motivation of this submission is to provide a reasonable baseline system for future research. We believe that the sub-field of non-autoregressive NMT cannot progress without controlled decoding speed evaluation, which is exactly what the shared task organizers provide.

Acknowledgements

This project received funding from the European Union's Horizon Europe Innovation programme under grant agreement no. 101070350 (HPLT). Our work has been using data provided by the LINDAT/CLARIAH-CZ Research Infrastructure, supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2018101).

[1] https://github.com/baidu-research/warp-ctc

References

Alexandra Birch, Andrew Finch, Minh-Thang Luong, Graham Neubig, and Yusuke Oda. 2018. Findings of the second workshop on neural machine translation and generation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 1–10, Melbourne, Australia. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Pinzhen Chen, Jindřich Helcl, Ulrich Germann, Laurie Burchell, Nikolay Bogoychev, Antonio Valerio Miceli Barone, Jonas Waldendorf, Alexandra Birch, and Kenneth Heafield. 2021. The University of Edinburgh's English-German and English-Hausa submissions to the WMT21 news translation task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 104–109, Online. Association for Computational Linguistics.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 369–376, Pittsburgh, PA, USA. JMLR.org.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018*, Vancouver, BC, Canada.

Hiroaki Hayashi, Yusuke Oda, Alexandra Birch, Ioannis Konstas, Andrew Finch, Minh-Thang Luong, Graham Neubig, and Katsuhito Sudoh. 2019. Findings of the third workshop on neural generation and translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 1–14, Hong Kong. Association for Computational Linguistics.

Kenneth Heafield, Hiroaki Hayashi, Yusuke Oda, Ioannis Konstas, Andrew Finch, Graham Neubig, Xian Li, and Alexandra Birch. 2020. Findings of the fourth workshop on neural generation and translation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 1–9, Online. Association for Computational Linguistics.

Kenneth Heafield, Qianqian Zhu, and Roman Grundkiewicz. 2021. Findings of the WMT 2021 shared task on efficient translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 639–651, Online. Association for Computational Linguistics.

Jindřich Helcl, Barry Haddow, and Alexandra Birch. 2022. Non-autoregressive machine translation: It's not as fast as it seems. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1780–1790, Seattle, United States. Association for Computational Linguistics.

Jindřich Helcl. 2022. *Non-Autoregressive Neural Machine Translation*. Ph.D. thesis, Charles University.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium. Association for Computational Linguistics.

Jindřich Libovický and Jindřich Helcl. 2018. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3016–3021, Brussels, Belgium. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

The RoyalFlush System for the WMT 2022 Efficiency Task

Bo Qin[1], **Aixin Jia**[1], **Qiang Wang**[2,1],
Jianning Lu[1], **Shuqin Pan**[1], **Haibo Wang**[1], **Ming Chen**[1*]
[1]RoyalFlush AI Research Institute, Hangzhou, China
[2]Zhejiang University, Hangzhou, China
{qinbo, jiaaixin, wangqiang3}@myhexin.com
{lujianning, panshuqin, wanghaibo3, chenming}@myhexin.com

Abstract

This paper describes the submission of the ROYALFLUSH neural machine translation system for the WMT 2022 translation efficiency task. Unlike the commonly used autoregressive translation system, we adopted a two-stage translation paradigm called Hybrid Regression Translation (HRT) to combine the advantages of autoregressive and non-autoregressive translation. Specifically, HRT first autoregressively generates a discontinuous sequence (e.g., make a prediction every k tokens, $k > 1$) and then fills in all previously skipped tokens at once in a non-autoregressive manner. Thus, we can easily trade off the translation quality and speed by adjusting k. In addition, by integrating other modeling techniques (e.g., sequence-level knowledge distillation and deep-encoder-shallow-decoder layer allocation strategy) and a mass of engineering efforts, HRT improves 80% inference speed and achieves equivalent translation performance with the same-capacity AT counterpart. Our fastest system reaches 6k+ words/second on the GPU latency setting, estimated to be about 3.1x faster than the last year's winner.

1 Introduction

Large-scale transformer models have made impressive progress in past WMT translation tasks, but it is still challenging for practical model deployment due to time-consuming inference speed (Wang et al., 2018b; Li et al., 2019). To build a fast and accurate machine translation system, participants in past WMT efficiency tasks developed and validated many efficient techniques, such as knowledge distillation (Hinton et al., 2015; Kim and Rush, 2016), light network architecture (Kasai et al., 2020), quantization (Lin et al., 2020) etc. We noticed that all the above efforts are aimed at autoregressive translation (AT) models. In contrast, other translation paradigms, like non-autoregressive translation (NAT) (Gu et al., 2017) or semi-autoregressive translation (SAT) (Wang et al., 2018a) etc., have not been well studied.

In this participation, we restrict ourselves to the GPU latency track and attempt to investigate the potential of non-standard translation paradigms. However, replicating the vanilla non-autoregressive or semi-autoregressive models degrades the translation quality severely in our preliminary experiments. To this end, we explore hybrid-regressive translation (HRT), the two-stage translation prototype, to better combine the advantages of autoregressive and non-autoregressive translation (Wang et al., 2021b). Specifically, HRT first uses an autoregressive decoder to generate a discontinuous target sequence with the interval k ($k > 1$). Then, HRT fills the remaining slots at once with a non-autoregressive decoder. The two decoders share the same parameters without adding additional ones. Thus, HRT can easily trade-off between translation quality and speed by adjusting k [1]. Please see Table 1 for the comparison between different translation paradigms.

In addition to the change of translation paradigm, we have also made a mass of other optimizations. We use the widely used sequence-level knowledge distillation (Kim and Rush, 2016) and deep-encoder-shallow-decoder layer allocation strategy (Kasai et al., 2020) to learn effective compact models. Moreover, on the engineering side, we customized an efficient implementation of GPU memory reuse and kernel fusion for HRT following LightSeq (Wang et al., 2021c).

Putting all the efforts together, our HRT model achieves almost equivalent BLEU scores to the corresponding AT counterparts while improving the inference speed by about 80%. Our best-BLEU

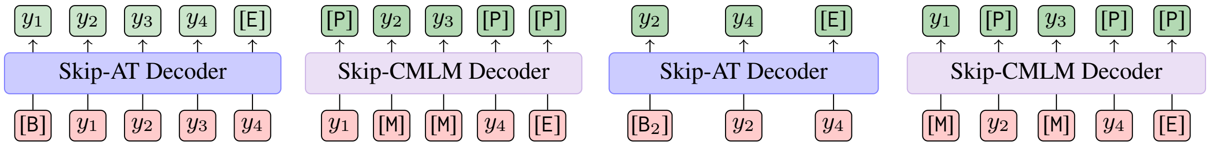
[*] Corresponding author.

[1] A larger k implies that fewer autoregressive decoding steps are required, resulting in faster inference speed but lower translation quality.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 671 - 676
December 7-8, 2022 ©2022 Association for Computational Linguistics

Source	__The __Next __Big __Labor __ Strike __Hit s __Oregon
AT	__Der → __nächste → __große → __Arbeits → streik → __trifft → __Oregon → [EOS]
SAT	__Der __nächste → __große __Arbeits → streik __trifft → __Oregon [EOS]
NAT	**__Der** **__nächste** **__große** **__Arbeits** **streik** **__trifft** **__Oregon** **[EOS]**
HRT (Stage I)	__nächste → __Arbeits → __trifft → [EOS]
HRT (Stage II)	**__Der** __nächste **__große** __Arbeits **streik** __trifft **__Oregon** [EOS]

Table 1: Illustrations of different translation paradigms. __ is the special symbol for whitespace in sentencepiece. → denotes an autoregressive decoding step. **Blue** denotes that the token is generated in non-autoregressive way.



(a) AT (b) CMLM (c) SKIP-AT (d) SKIP-CMLM

Figure 1: Examples of training samples for four tasks, in which (a) and (b) are auxiliary tasks and (c) and (d) are primary tasks. For the sake of clarity, we omit the source sequence. [B]/[E]/[P]/[M] represents the special token for [BOS]/[EOS]/[PAD]/[MASK], respectively. [B$_2$] is the [BOS] for k=2. Loss at [P] is ignored.

model drops an average of 0.9 BLEU points compared to the teacher model, which ensembles four transformer-big models. Moreover, our fastest model decodes 6k+ source words per second, estimated to be 3.1x faster than the winner in last year [2].

2 Hybrid-regressive translation

One of the most important highlights is the introduction of the newly proposed two-stage translation prototype——HRT. In this section, we will detail the model, training, and decoding of HRT.

2.1 Model

HRT consists of three components: encoder, Skip-AT decoder (for stage I), and Skip-CMLM decoder (for stage II). All components adopt the Transformer architecture (Vaswani et al., 2017). The two decoders have the same network structure, and we share them to make the parameter size of HRT the same as the vanilla Transformer. The only difference between the two decoders lies in the masking pattern in self-attention: The Skip-AT decoder masks future tokens to guarantee strict left-to-right generation like an autoregressive Transformer (Vaswani et al., 2017). In contrast, the Skip-CMLM decoder eliminates it to leverage the bi-directional context like the standard conditional masked language model (CMLM) (Ghazvininejad et al., 2019). We note that there is no specific target length prediction module in HRT because HRT can obtain the translation length as the by-product of the Skip-AT decoder: N_{nat}=$k \times N_{at}$, where N_{at} is the sequence length produced by Skip-AT.

2.2 Training

Multi-task framework. We learn HRT through joint training of four tasks, including two primary tasks (SKIP-AT, SKIP-CMLM) and two auxiliary tasks (AT, CMLM). All tasks use cross-entropy as the training objective. Figure 1 illustrates the differences in training samples among these tasks. It should be noted that, compared with AT, SKIP-AT shrinks the sequence length from N to N/k, whereas the token positions follow the original sequence. For example, in Figure 1 (c), the position of Skip-AT input ([B$_2$], y_2, y_4) is (0, 2, 4) instead of (0, 1, 2). Involving auxiliary tasks is necessary because the two primary tasks cannot fully leverage all tokens in the sequence due to the fixed k. For example, in Figure 1 (c) and (d), y_1 and y_3 have no chance to be learned as the decoder input of either SKIP-AT or SKIP-CMLM.

Curriculum learning. To ensure that the model is not overly biased towards auxiliary tasks, we propose gradually transferring the training tasks from

[2] We obtained the best decoding speed in last year's competition according to Heafield et al. (2021): The fastest system *2.12_1.micro.rowcol-0.5* decodes 19,951,184 space-separated words in 13665 seconds. Therefore, we estimated its inference speed is 1460 words/second. We note that the acceleration ratio is not an accurate value because we use slightly different computation devices and test data to measure the speed.

auxiliary tasks to primary tasks through curriculum learning (Bengio et al., 2009). More concretely, given a batch of original sentence pairs $\mathcal{B}$, and the proportion of primary tasks in $\mathcal{B}$ is p_k, we start with p_k=0 and construct the training samples of AT and CMLM for all pairs. Then we gradually increase p_k to introduce more learning signals for SKIP-AT and SKIP-CMLM until p_k=1. In implementation, we schedule p_k by:

$$p_k = (t/T)^{\lambda}, \tag{1}$$

where t and T are the current and total training steps. λ is a hyperparameter, and we use λ=1 to increase p_k linearly for all experiments.

2.3 Decoding

HRT adopts two-stage generation strategy: In the first stage, the Skip-AT decoder starts from [BOS$_k$] to autoregressively generate a discontinuous target sequence $\hat{\boldsymbol{y}}_{at} = (z_1, z_2, \ldots, z_m)$ with chunk size k until meeting [EOS]. Then we construct the input of Skip-CMLM decoder $\boldsymbol{y}_{nat}$ by appending $k-1$ [MASK]s before every z_i. The final translation is generated by replacing all [MASK]s with the predicted tokens by the Skip-CMLM decoder with one iteration. If there are multiple [EOS]s existing, we truncate to the first [EOS]. Note that the beam size b$_{at}$ in Skip-AT can be different from the beam size b$_{nat}$ in Skip-CMLM as long as st. b$_{at} \geq$ b$_{nat}$: We only feed the top b$_{nat}$ Skip-AT hypothesis to Skip-CMLM decoder. Finally, we choose the translation hypothesis with the highest score S($\hat{y}$) by:

$$\underbrace{\sum_{i=1}^{m} \log P(z_i|\boldsymbol{x}, \boldsymbol{z}_{<i})}_{\text{Skip-AT score}} + \underbrace{\sum_{i=0}^{m-1}\sum_{j=1}^{k-1} \log P(\hat{y}_{i\times k+j}|\boldsymbol{x}, \boldsymbol{y}_{nat})}_{\text{Skip-CMLM score}} \tag{2}$$

where z_i=$\hat{y}_{i\times k}$.

3 Optimization

Sequence-level knowledge distillation. Overall, we use the teacher-student framework via sequence-level knowledge distillation (SEQKD) to learn our small HRT model (Kim and Rush, 2016). Specifically, the ensemble of provided four transformer-big models is our teacher, whose beam search results are used as our distillation data. There are 320M official distillation data composed of 80M parallel and 240M monolingual datasets. We directly use the distillation data without further data cleaning. We use the same sentencepiece vocabulary as the teacher model to encode the text.

Encoder	Newstest19	Newstest20	WPS
6	43.8	32.6	4.0k
12	45.6	33.9	3.8k
20	45.9	34.4	3.5k

Table 2: SacreBLEU and inference speed against the number of encoder layers in HRT with a single-layer decoder. All HRT models are trained with k=2. WPS refers to source words per second, measured by the average five runs with a batch size of 1. Unless otherwise stated, we measure WPS on Newstest20.

b$_{at}$	b$_{nat}$	Newstest19	Speedup
5	5	**45.9**	ref.
5	1	45.8	1.05x
1	1	45.6	**1.33x**

Table 3: Effects of different settings of beam size in HRT.

Deep-encoder-shallow-decoder architecture. Using deep-encoder-shallow-decoder network architecture has been widely validated effectiveness for transformer-based NMT systems (Wang et al., 2021a; Kasai et al., 2020). Our HRT also follows this guidance by using only one decoder layer. We use the pre-norm transformer following Wang et al. (2019) to learn deep encoder well. Intuitively, the single-layer decoder may be insufficient for HRT because the decoder is responsible for both autoregressive and non-autoregressive generation. However, as shown in Table 2, we found that HRT enjoys the deep-encoder-shallow-decoder architecture. For example, compared to HRT_E6D1 [3], HRT_E12D1 and HRT_E20D1 improve +1.6/+2.0 BLEU score points on average, while the inference speed decreases by 5% and 12.5%. Therefore, we mainly investigate HRT with a 12-layer and 20-layer encoder due to the high BLEU scores.

Fully greedy search. Prior work has validated that greedy search is sufficient for the autoregressive distilled model to work well (Kim and Rush, 2016). Since HRT refers to two beam sizes (b$_{at}$ and b$_{nat}$), we test three settings as shown in Table 3. It can be seen that using b$_{at}$=1 and b$_{nat}$=1 only decreases BLEU slightly but accelerates a 30%+ faster than that of b$_{at}$=5 and b$_{nat}$=5. Unless otherwise stated, we use b$_{at}$=1 and b$_{nat}$=1 in the following experiments.

[3] We use HRT_E{#1}D{#2} to denote the HRT model with {#1}-layer encoder and {#2}-layer decoder.

Model	Param.	Newstest19		Newstest20		Average		WPS
		BLEU	**COMET**	**BLEU**	**COMET**	**BLEU**	**COMET**	
Teacher (four transformer-big)	4×209.1M	47.1	-	35.0	-	41.1	-	-
WMT21 fastest (Behnke et al., 2021)	9.0M	-	-	33.3	-	-	-	1.5k*
AT_E6D1	39.5M	45.1	0.551	33.9	0.469	39.5	0.510	2.2k
AT_E12D1	58.4M	45.4	0.572	34.1	0.489	39.8	0.530	2.1k
AT_E20D1	83.6M	45.9	0.581	34.6	0.502	40.3	0.541	1.9k
HRT_E12D1 (k=2)	58.4M	45.6	0.547	33.9	0.454	39.8	0.500	3.8k
HRT_E12D1 (k=3)	58.4M	45.0	0.503	33.6	0.377	39.3	0.440	4.9k
HRT_E12D1 (k=4)	58.4M	44.1	0.432	32.9	0.267	38.5	0.350	**6.1k**
HRT_E20D1 (k=2)	83.6M	45.9	0.561	34.4	0.472	**40.2**	**0.517**	3.5k
HRT_E20D1 (k=3)	83.6M	45.3	0.524	34.0	0.406	39.7	0.465	4.5k
HRT_E20D1 (k=4)	83.6M	44.2	0.435	33.3	0.283	38.8	0.360	5.4k

Table 4: Compare different model variants with regard to SacreBLEU (Post, 2018), COMET (Rei et al., 2020) and inference speed. $*$ denotes the number is not exactly comparable due to the difference in test data and GPU.

Maximum sequence length. We predefine the maximum source/target sequence length L as 200. Here the length is calculated based on the results of sentencepiece. Once the sequence length exceeds L, we truncate the source/target sequence. For HRT, we let the maximum decoding length in the Skip-AT stage as L/k. In this way, the maximum target length in the Skip-CMLM stage can be guaranteed not beyond L.

GPU memory reuse. Since we only participate in the GPU latency track, given the predefined maximum sequence length L, we can estimate the maximum GPU memory buffer used in the encoder, autoregressive decoder, and non-autoregressive decoder in advance, respectively. Then we only allocate the maximum buffer size among them because these three processes are memory-independent. This memory-reuse method helps us reduce our footprints and avoid frequent memory applications and releases.

Kernel fusion. Too many fine-grained kernel functions make modern GPU inefficient due to kernel launching overhead and frequent memory I/O addressing (Wang et al., 2021c; Wu et al., 2021). We follow the good implementation in LightSeq and use the general matrix multiply (GEMM) provided by cuBLAS as much as possible, with some custom kernel functions. Please refer to Wang et al. (2021c) for details.

FP16 inference. We also use the 16-bit floating-point to utilize modern GPU hardware efficiently. Previous study (Wang et al., 2021a) shows that FP16 can bring significant acceleration in batch decoding. In contrast, in our GPU latency task, we only observed about 15% speedup due to the smaller computational burden, as shown in Table 5.

Model	FP16	WPS
HRT_E12D1 (k=2)	no	3.3k
HRT_E12D1 (k=2)	yes	3.8k

Table 5: The effect of FP16 on HRT model in GPU latency task.

Docker submission. We use multistage builds to reduce the docker image size. Specifically, we first use static compilation to build our executable program with CUDA 11.2. Then we add the built result and model into the 11.2.0-base-centos7 docker. The model disk size is compressed by *xz* compression toolkit.

4 Experimental Results

Setup. We mainly compared HRT to the standard autoregressive baselines in Table 4. All models adopt transformer-base setting (Vaswani et al., 2017): d=512, d_{ff}=2048, $head$=8. We validated the following model variants:

- **AT:** We train three autoregressive baselines with the number of encoder layers of 6, 12, and 20, denoted as AT_E6D1, AT_E12D1, and AT_E20D1, respectively. All AT models are trained from scratch for 300k steps.

- **HRT:** HRT models are fine-tuned based on the pre-trained AT counterparts for 300k steps. We also try different chunk sizes $k \in \{2, 3, 4\}$ to trade off the translation quality and inference speed.

Other training hyper-parameters are the same as Wang et al. (2019). We ran all experiments on 8 GeForce 3090 GPUs. For decoding, the length penalty is 0.6, and the batch size is 1. We report the detokenized SacreBLEU score with the same signature as the teacher. Besides, we also follow Helcl et al. (2022)'s suggestion to provide COMET score (Rei et al., 2020) for the evaluation of non-autoregressive translation.

Translation quality. First, we can see that a deeper encoder improves about 0.5 BLEU points across the board. When using k=2 for HRT, both 12-layer and 20-layer HRT models have almost equivalent BLEU scores to that of AT counterparts. Our best HRT model HRT_E20D1 with k=2 only drops an average of 0.9 BLEU points than the teacher using model ensemble. However, in line with Helcl et al. (2022), we find that even when BLEU scores are close, HRT's COMET scores are significantly lower than those of AT, e.g., AT_E20D1 vs. HRT_E20D1 (k=2). Nevertheless, HRT_E20D1 (k=2) still achieves higher BLEU and COMET than AT_E6D1 with 60% acceleration.

Translation speed. We estimated the inference speed of the fastest system last year according to the data in Heafield et al. (2021). Supposing ignoring the difference in test data, our AT baselines run about 40%+ faster than it. It indicates that our AT engine is a strong baseline. Even so, we can see that both 12-layer and 20-layer HRT with k=2 achieve approximated 80% acceleration than AT without BLEU drop. Moreover, larger k further reduces the autoregressive decoding steps: Our fastest model, HRT_E12D1 (k=4), decodes 6k+ source words/second, which is 3.1 times faster than the fastest system last year.

5 Conclusion

This paper presented the RoyalFlush system to the GPU latency track of the WMT 2022 translation efficiency task. We proposed hybrid-regressive translation, a novel two-stage prototype to replace conventional autoregressive translation. With a lot of development optimization, we showed that our HRT with a chunk size of 2 achieves equivalent translation performance to the AT counterpart while accelerating 80% inference speed. By increasing HRT's chunk size, our system can further speed up 60% to 6k+ words/second, estimated to be about 3.1 times faster than the fastest system in last year's competition.

References

Maximiliana Behnke, Nikolay Bogoychev, Alham Fikri Aji, Kenneth Heafield, Graeme Nail, Qianqian Zhu, Svetlana Tchistiakova, Jelmer van der Linde, Pinzhen Chen, Sidharth Kashyap, and Roman Grundkiewicz. 2021. Efficient machine translation with model pruning and quantization. In *Proceedings of the Sixth Conference on Machine Translation*, pages 775–780, Online. Association for Computational Linguistics.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6111–6120, Hong Kong, China. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2017. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*.

Kenneth Heafield, Qianqian Zhu, and Roman Grundkiewicz. 2021. Findings of the WMT 2021 shared task on efficient translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 639–651, Online. Association for Computational Linguistics.

Jindřich Helcl, Barry Haddow, and Alexandra Birch. 2022. Non-autoregressive machine translation: It's not as fast as it seems. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1780–1790, Seattle, United States. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A. Smith. 2020. Deep encoder, shallow decoder: Reevaluating the speed-quality tradeoff in machine translation. *arXiv preprint arXiv:2006.10369*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The NiuTrans machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266, Florence, Italy. Association for Computational Linguistics.

Ye Lin, Yanyang Li, Tengbo Liu, Tong Xiao, Tongran Liu, and Jingbo Zhu. 2020. Towards fully 8-bit integer inference for the transformer model. *ArXiv*, abs/2009.08034.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Chenglong Wang, Chi Hu, Yongyu Mu, Zhongxiang Yan, Siming Wu, Yimin Hu, Hang Cao, Bei Li, Ye Lin, Tong Xiao, and Jingbo Zhu. 2021a. The NiuTrans system for the WMT 2021 efficiency task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 787–794, Online. Association for Computational Linguistics.

Chunqi Wang, Ji Zhang, and Haiqing Chen. 2018a. Semi-autoregressive neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 479–488.

Qiang Wang, Bei Li, Jiqiang Liu, Bojian Jiang, Zheyang Zhang, Yinqiao Li, Ye Lin, Tong Xiao, and Jingbo Zhu. 2018b. The NiuTrans machine translation system for WMT18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 528–534, Belgium, Brussels. Association for Computational Linguistics.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822, Florence, Italy.

Qiang Wang, Heng Yu, Shaohui Kuang, and Weihua Luo. 2021b. Hybrid-regressive neural machine translation.

Xiaohui Wang, Ying Xiong, Yang Wei, Mingxuan Wang, and Lei Li. 2021c. LightSeq: A high performance inference library for transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 113–120, Online. Association for Computational Linguistics.

Kaixin Wu, Bojie Hu, and Qi Ju. 2021. TenTrans high-performance inference toolkit for WMT2021 efficiency task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 795–798, Online. Association for Computational Linguistics.

HW-TSC's Submissions to the WMT22 Efficiency Task

Hengchao Shang[1], Ting Hu[2], Daimeng Wei[1], Zongyao Li[1],
Xianzhi Yu[2], Jianfei Feng[2], Jinlong Yang[1], Zhiqiang Rao[1], Ting Zhu[1],
Zhengzhe Yu[1], Lizhi Lei[1], Shimin Tao[1], Hao Yang[1], Ying Qin[1]
[1]Huawei Translation Service Center, Beijing, China
[2]Huawei Noah's Ark Lab, Hong Kong, China
{shanghengchao,huting35,weidaimeng,lizongyao,yuxianzhi,
fengjianfei1,yangjinlong7,raozhiqiang,zhuting20,
yuzhengzhe,leilizhi,taoshimin,yanghao30,qinying}@huawei.com

Abstract

This paper presents the submissions of Huawei Translation Services Center (HW-TSC) to WMT 2022 Efficiency Shared Task. For this year's task, we still apply sentence-level distillation strategy to train small models with different configurations. Then, we integrate the average attention mechanism into the lightweight RNN model to pursue more efficient decoding. We add a retrain step to our 8-bit and 4-bit models to achieve a balance between model size and translation quality. We still use Huawei Noah's Bolt[1] for INT8 inference and 4-bit storage. With Bolt's support for batch inference and multi-core parallel computing, we finally submit models with different configurations to the CPU latency and throughput tracks to explore the Pareto frontiers.

1 Introduction

Transformer and its variants (Vaswani et al., 2017; Shaw et al., 2018; So et al., 2019; Dehghani et al., 2019) have become benchmark models for machine translation. A lot of innovations and engineering optimizations (Tay et al., 2020) in this area are based on Transformer. However, with the increase of bilingual and monolingual data sizes used for training, the size of the model expands and the requirement of computing ability become higher. Taking T5 (Raffel et al., 2020), GPT3 (Brown et al., 2020) and a series of subsequent large models (Fedus et al., 2021; Smith et al., 2022) as examples, although they have achieved very good performances, it is still difficult for ordinary practitioners to reproduce or use these models for research and industry application. Especially in scenarios where hardware capability is limited, models that balance size, quality and power consumption is urgently needed. The WMT Efficiency task is performed under such constraints.

In this year's task, we still focus on CPU inference optimization and participate in CPU latency and multi-core throughput tracks.

We employ knowledge distillation (Hinton et al., 2015) to train small models. The teacher models and distillation data come from official website. We only perform simple data cleaning, and all of our experiments are conducted based on fariseq (Ott et al., 2019).

Deep encoder and shallow deocder models can balance quality and inference speed (Wang et al., 2019). We follow this configuration for pursuing extreme efficient decoding. Inspired by SRU++ (Lei, 2021) and AAN (Zhang et al., 2018), we integrate the average attention mechanism with a lightweight RNN for more efficient decoding.

We retrain our 8-bit quantisation model (Jacob et al., 2018), then compare its result with that of direct post-quantization (Sung et al., 2015) model. We finally find that in the distillation scenario, the difference between the two is not obvious. We apply 4-bit storage to obtain an extremely small model size. Although our training and inference strategies ensure basically the same model quality, the gap in overall quality is large and the model needs to be further optimized.

We still use Huawei Noah's Bolt as the inference library. This year, we implement batch inference and parallel computing on multi-core CPUs for the throughput track.

Finally, after performing some necessary engineering optimizations, we submit four models with different configurations to explore the Pareto frontiers.

2 Teacher to Student Knowledge Distillation

2.1 Data Process

The task is to translate English to German following the constrained news task from WMT 2021.

[1] https://github.com/huawei-noah/bolt

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 677 - 681
December 7-8, 2022 ©2022 Association for Computational Linguistics

The teacher model, as well as bilingual data and distillation data used in this task are provided by the organizer. It makes everyone on the same start line in the distillation experiment, avoiding the quality difference due to different teacher models. We download the data and find it pretty much the same as the data we used in the task last year. According to our distillation experiment last year, keeping the ratio of bilingual data and distillation data as 1:2 can ensure that the student model inherits the knowledge of the teacher model well. Except for the generation of distillation data, other processing strategies are the same as last year's. For details, please refer to our previous task report (Shang et al., 2021).

2.2 Vocabulary

We build a joint subword segmentation model from real parallel data using SentencePiece (Kudo and Richardson, 2018) as last year. The vocabulary size is set to 25k tokens.

2.3 Model Structure

The autoregressive module is based on the self-attention in the Transformer decoder layer. The decoding complexity increases as the decoding length increases. Therefore, special processing is required if we want to pursue extreme decoding performance. The commonly used strategy is to replace it with a fixed computational cost module, such as LSTM, other RNN variants (Lei et al., 2018), or AAN. These modules use a global cell to store sentence-level information and perform the same cell update actions as each token is decoded without relying on the decoded sequence.

SRU++ (Lei, 2021) further replaces the heavyweight multiplication operation outside the cell with a self-attention component to improve the representation ability of the model. We use the AAN to replace the standard self-attention module for faster decoding while ensuring the expression ability of the model. We call it the AASRU model.

The calculation formula in the cell is as follows:

$$\begin{aligned}
f[t] &= \sigma(U[t,0]) + V \odot C[t-1] + b) \\
r[t] &= \sigma(U[t,1]) + V^{'} \odot C[t-1] + b^{'}) \\
c[t] &= f[t] \odot C[t-1] + (1 - f[t]) \odot C[t,2] \\
h[t] &= r[t] \odot C[t] + (1 - r[t]) \odot x[t]
\end{aligned}$$

The formula for calculating U is as follows:

$$Q = W^q X^T$$

$$\begin{aligned}
V &= W^v X^T \\
A^T &= AVERAGE(V^T) \\
U^T &= W^o layernorm(Q + A)
\end{aligned}$$

where W^q and $W^v \in R^{d^{'} \times d}$, $W^o \in R^{3d \times d^{'}}$, d is the hidden state size, and $d^{'}$ is the attention dimension. The σ is the sigmode function, and $\odot$ is the element-wise multiplication, t refer to the time step, v and b are parameter vectors to be learnt during training, c and h are the cell states and the hidden states in RNN.

2.4 Training

Our distillation experiments are based on fairseq. We implement the AASRU module by referring to the open-source transformer-aan [2]. Also, we do not use regularization techniques such as dropout and label smoothing. All our models are trained using 8 Nvidia Tesla V100 for about two days. The maximum number of tokens vary from 4096 to 10240 according to the model size, as we try to keep the maximum GPU memory usage the same.

After that, we retrain our 8-bit quantization model, constrain all Linear and Matual operator's inputs to the interval [-1, 1], add quantization and inverse quantization operators to the model graph. The retrain is performed after the base model has been trained for 200K steps.

We compare the results of retrain and post-quantization on the Base.12 model, and find almost no difference in performances of the two models under the current distillation experiment setting. Therefore, we submit the post-quantized models.

Next, we apply 4-bit storage models to pursue extreme model sizes. In order to achieve better translation, we add retrain and verify the consistency between training and inference. The translation quality obtained via Bolt inference and training respectively is almost the same. However, the overall quality of our model declines greatly, requiring further optimization.

2.5 Evaluation

We still use WMT 2019 and 2020 News Task test sets to measure our models with SacreBLEU (Post, 2018) this year. We perform a simple post-processing (normalize the punctuation) on the German translations, so the BLEU scores are slightly higher than the officially provided one.

[2] https://github.com/bzhangGo/transformer-aan

Model	**Emb.**	**FFN**	**Head**	**Depth**	**Params(M)**	**Size(MB)**	**wmt19**	**wmt20**
Teacher*4	1024	4096	16	6/6	200	800	47.08	36.29
Base.12	512	2048	8	12/1	53	210	45.75	35.30
Base.12 + 8-bit	512	2048	8	12/1	53	210	45.89	35.20
Base.6	512	2048	8	6/1	35	140	44.78	34.59
Small.12	384	1536	6	12/1	33	132	45.03	34.89
Small.9	384	1536	6	9/1	28	112	44.62	34.40
Small.6	384	1536	6	6/1	22	88	43.80	34.26
Tiny.12	256	1024	4	12/1	17	68	43.84	33.62
Tiny.6	256	1024	4	6/1	13	52	42.15	32.27
Tiny.6 + 4-bit	256	1024	4	6/1	13	52	34.75	26.30

Table 1: Results of Distillation Training. 8-bit and 4-bit refer to retraining.

Overall, the results of our distillation experiments are within our expectations. The Baseline model has about 25% parameters as the teacher model, and its performance is attenuated by about 1.5 BLEU. The 8-bit retraining model is basically the same as the direct training one. However, we observe over 5.0 BLEU decrease on our Tiny.6 model after adding 4-bit storage. The reason may be that we treat every parameter the same way, including embedding. As a result, more training tricks and experiments are required in the future.

We also analyze the effect of the encoder's height and width on the model. Comparing Base.6 to Small.12, and Small.6 to Tiny.12, we find that deeper networks almost have equal or better quality even with less parameters except for Tiny.12's 2020 test result.

Under the same height setting, models with different widths also perform differently. A wider model seems to perform better. Comparing 12-layer and 6-layer models, we observe less than 1 BLEU difference under the base setting, less than 1.2 BLEU difference under the small setting (and only 0.7 BLEU difference on the WMT20 test set), and only about 1.7 BLEU difference under the tiny setting. Wider encoder means more parameters and probably better quality.

Based on the above analysis and the quality gap between the models, We finally decide to submit four models including Base.12, Small.9, Tiny.12, and Tiny.6 to explore the Pareto frontiers better.

3 Inference Optimizations

We use Bolt acceleration library as CPU optimization backend to build the high-performance translation engine. Bolt has a standalone C++ runtime, therefore it can perform fast inference without any third-party dependencies. We use Bolt v1.4.0, which will be available in October 2022.

3.1 8-bit Quantization

We still apply the post-training quantization method this year. All parameters of the model except the bias are quantized to 8-bit intergers by absolute maximum quantization. All GEMM operations in the attention layer are in 8-bit and well optimized by Intel VNNI instructions, but the layernorm and softmax computations are back off to FP32.

3.2 4-bit Storage

For this year's submission, we employ 4-bit storage to achieve almost 8x model compression. With 4-bit storage, all parameters have to be converted to 8-bit integers for calculation because of the hardware limitations, so there is no performance advantage compared with 8-bit storage.

3.3 Batch and Thread

For the throughput track, we support batch inference and merge multiple matrix calculations in attention layer. Our experiments show an end-to-end speedup of up to 20% on a single core. To further increase the throughput, we divide the input text into specified sizes and assign them to multiple CPU cores for parallel computation. The input text is sorted first to prevent the performance waste due to the difference of data lengths within the batch. In the submitted systems, we uniformly set the batch size to 4.

3.4 Other Strategies

We apply some other commonly used strategies such as greedy decoding, caching and shortlist,

Model	Precious	Size	WPS	BLEU
Teacher	FP32	2000	-	36.29
Base.12	FP32	212	237	35.30
	INT8	53	815	35.20
+retrain	INT8	53	815	35.19
Small.9	FP32	112	468	34.40
	INT8	28	1129	34.29
Tiny.12	FP32	68	759	33.62
	INT8	17	1693	33.39
Tiny.6	FP32	52	996	32.27
	INT8	13	2001	31.92
+int4	INT8	6.5	1989	26.10

Table 2: Optimization results. The test set is WMT 2020 News test. The unit of size is MB. WPS refers to the source side. The test environment is Intel(R) Xeon(R) Gold 6278C CPU @ 2.60GH. We submit four models: Base.12, Small.9, Tiny.12 and Tiny.6 and the final Tiny.6+int4

which can improve the model decoding efficiency to a certain extent. Details can be found in our last year's report.

4 Optimization Results

Our final optimization results are shown in Table 2. We find that the inference speed of our models is significantly improved through int8 inference, and the overall improvement is 2-3 times that of FP32, which is basically the same as last year's results.

By analyzing the results of our comparative experiments on Base.12, we find that BLEU is only slightly decreased when we directly use the post-quantization inference version. So there is not much room left when optimizing the performance of models that employ retraining. The reason may be the limited diversity of the model under the distillation setting. The post-quantization model basically meet our requirement on quality.

We additionally employ 4-bit storage on the Tiny.6 model for pursuing extreme model size. After retraining, we successfully compress the model to almost 1/8 of the original size, and maintain a high degree of consistency between training (26.30 BLEU) and inference (26.10 BLEU) with slightly BLEU score decrease. We also submit the model for evaluation.

When preparing the model for the throughput track, we need to set the batch size for batch translation. We compare the impact of different batch sizes on throughput in detail using our Base.12 model. Results are shown in Table 3. When the batch size exceeds 3, the improvement becomes insignificant. Considering that the hardware used in the task may be different from our test environment, we set the batch size to 4 for all of our submissions for convenience.

Batch Size	BLEU	Costs	WPS
Base.12	35.30	-	-
1	35.20	53	815
2	35.23	45	978
3	35.17	44	1000
4	35.22	44	1000
8	35.38	45	978
16	35.26	45	978

Table 3: The effect of batch size on throughput. WPS refers to the source side.

5 Submitted Docker Images

Due to the simple runtime environment of Bolt, we can choose a very basic image to run our system. We still apply the ubuntu:18.04. Our inference project is inherited from last year's, adding support of batch inference and using a thread pool to run models in parallel on multiple CPU cores. Following the task requirements, our startup script is /run.sh. Our model is stored in the /model directory, which contains the converted Bolt model, vocabulary, and shortlist files. The compressed file is provided.

Our largest model volume is around 50M, and the base image volume is around 60M. The space occupied by our inference project is almost negligible, so the final image we submitted after compression still does not exceed 70M, and the smallest one is about 35M.

6 Concolusion

In this year's task, we follow some strategies from last year, including data processing, basic distillation training, etc. In addition, we explore a new and more efficient decoding structure, AASRU, this year, which reduces the amount of computation while maintaining quality. We add 8-bit and 4-bit retrain to distillation training, and verify the consistency of training and inference. Regarding engineering, we add the relevant features of Batch inference and multi-core parallel computing, and finally submit several models with balanced quality and speed for CPU latency and throughput tracks.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. 2019. Universal transformers.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71.

Tao Lei. 2021. When attention meets fast recurrence: Training language models with reduced compute. *arXiv preprint arXiv:2102.12459*.

Tao Lei, Yu Zhang, Sida I. Wang, Hui Dai, and Yoav Artzi. 2018. Simple recurrent units for highly parallelizable recurrence.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Hengchao Shang, Ting Hu, Daimeng Wei, Zongyao Li, Jianfei Feng, Zhengzhe Yu, Jiaxin Guo, Shaojun Li, Lizhi Lei, Shimin Tao, et al. 2021. Hw-tsc's participation in the wmt 2021 efficiency shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 781–786.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.

David R. So, Chen Liang, and Quoc V. Le. 2019. The evolved transformer.

Wonyong Sung, Sungho Shin, and Kyuyeon Hwang. 2015. Resiliency of deep neural networks under quantization. *arXiv preprint arXiv:1511.06488*.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020. Efficient transformers: A survey.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Biao Zhang, Deyi Xiong, and Jinsong Su. 2018. Accelerating neural transformer via an average attention network. *arXiv preprint arXiv:1805.00631*.

IIT Bombay's WMT22 Automatic Post-Editing Shared Task Submission

Sourabh Deoghare and **Pushpak Bhattacharyya**
Computation for Indian Langauge Technology (CFILT)
IIT Bombay, India
{sourabhdeoghare, pb}@cse.iitb.ac.in

Abstract

This paper describes IIT Bombay's submission to the WMT-22 Automatic Post-Editing (APE) shared task for the English-Marathi (En-Mr) language pair. We follow the curriculum training strategy to train our APE system. First, we train an encoder-decoder model to perform translation from English to Marathi. Next, we add another encoder to the model and train the resulting *dual-encoder single-decoder* model for the APE task. This involves training the model using the synthetic APE data in multiple training stages and then fine-tuning it using the real APE data. We use the LaBSE technique to ensure the quality of the synthetic APE data. For data augmentation, along with using candidates obtained from an external machine translation (MT) system, we also use the phrase-level APE triplets generated using phrase table injection. As APE systems are prone to the problem of 'over-correction', we use a sentence-level quality estimation (QE) system to select the final output between an original translation and the corresponding output generated by the APE model. Our approach improves the TER and BLEU scores on the development set by -3.92 and +4.36 points, respectively. Also, the final results on the test set show that our APE system outperforms the baseline system by -3.49 TER points and +5.37 BLEU points.

1 Introduction

Automatic Post-Editing (APE) is a post-processing task in a Machine Translation (MT) workflow. It aims to automatically identify and correct errors in MT outputs (Chatterjee et al., 2020). Läubli et al. (2013) and Pal et al. (2016) show that APE systems have the potential to reduce human effort by automatically correcting repetitive translation errors.

The initial years of the WMT APE shared task focused on correcting errors in Statistical Machine Translation (SMT) translations, where participants explored various statistical and neural APE approaches (Bojar et al., 2017). Although neural APE approaches showed high potential for significantly improving the quality of SMT translations, these approaches faced challenges in improving translations obtained from relatively-more-robust neural machine translation (NMT) systems (Chatterjee et al., 2018). A possible reason for this could be that correcting a high-quality translation requires fewer edits, and therefore APE approaches need to be precise in identifying and in correcting the errors. Also, the neural APE approaches use large neural networks that require significant training data. APE training data consists of 'triplets' in the form of source sentence (*src*), its translation generated using an MT system (*mt*), and a human post-edited version of the translation (*pe*). Obtaining *pe* is an expensive task in terms of time and money; therefore, there is a lack of large APE datasets.

To deal with this problem, various data augmentation techniques have been proposed (Junczys-Dowmunt and Grundkiewicz, 2016; Negri et al., 2018; Lee et al., 2020b). Wang et al. (2020) used imitation learning to filter the APE data for tackling the distributional difference between real and synthetic APE data. Wei et al. (2020) augmented the APE training data with translations generated using a different MT system. Inspiring from the work of Sen et al. (2021), we augment the APE data by generating phrase-level APE triplets using SMT phrase tables. To ensure the quality of the synthetic data, we use the LaBSE technique (Feng et al., 2022) and filter low-quality triplets.

Another effective approach for dealing with the problem of data sparsity is transfer learning in which pre-trained models are adapted to the APE task (Lopes et al., 2019). An APE system needs to understand both the source and target languages to obtain joint encoding of *src* and *mt*. Therefore, Lee et al. (2020a) uses a cross-lingual language model instead of a monolingual one. Unlike

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 682 - 688
December 7-8, 2022 ©2022 Association for Computational Linguistics

these approaches, Wei et al. (2020); Sharma et al. (2021) use a pre-trained NMT model and adapts it to the APE task. Oh et al. (2021) has proposed the Curriculum Training Strategy (CTS) that gradually adapts pre-trained models to the APE task.

Although recent APE systems use a single encoder to encode both the source sentence and its translation (Oh et al., 2021; Lee et al., 2020a), we use separate encoders for encoding *src* and *mt* as English and Marathi do not share much vocabulary; and belong to different language families. We use IndicBERT (Kakwani et al., 2020) to initialize weights of our the *src* encoder and *mt* encoder. We train and fine-tune our models using the CTS over the good-quality APE data. The training data is also augmented with external MT candidates and phrase-level APE triplets. It is known that APE systems are prone to making unnecessary edits to translation output (Chatterjee et al., 2020). To mitigate this issue of over-correction, we use a sentence-level QE system to select the final output. When evaluated on the development set, our approach improves the TER (Snover et al., 2006) by -3.92 points and the BLEU (Papineni et al., 2002) by +4.98 points. Similarly, the final results on the test set show that our APE system outperforms the baseline system by -3.49 TER points and +5.37 BLEU points. We summarize the main features of our approach as follows:

- We use two separate encoders to generate representations for *src* and *mt*. We also use the IndicBERT language model to initialize the weights for both our encoders.

- We filter low-quality APE triplets from the synthetic data using LaBSE-based filtering.

- We divide the APE training step using CTS into two phases. We train the APE model in the first phase using out-of-domain synthetic APE data. In the next phase, we train the APE model using only the in-domain APE data.

- We follow two approaches for data augmentation: (1) As per the recent trend, we use external MT candidates. (2) We generate phrase-level APE triplets using SMT phrase tables.

- APE systems are prone to the problem of over-correction. Therefore, we use a sentence-QE system to select the final output between the APE output and the original translation.

2 Approach

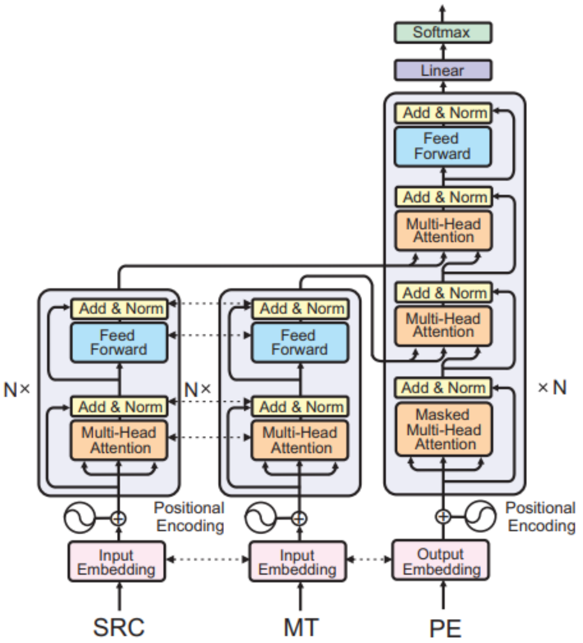


Figure 1: Dual-encoder Single Decoder Architecture. Dashed arrows represent tied parameters and common embedding matrices for encoders and decoder.(Junczys-Dowmunt and Grundkiewicz, 2018)

Our APE model is based on the transformer (Vaswani et al., 2017) architecture. Figure 1 shows the architecture of our APE model. In this section, we discuss the details of our approach.

2.1 Dual-Encoder Single-Decoder APE Model

The APE task is usually treated as an NMT-like task. Recent approaches use a single encoder to encode a source sentence and its translation (Oh et al., 2021; Lee et al., 2020a). Such an approach may work well when the source and target languages share the vocabulary (Kanojia et al., 2021). However, for English and Marathi, there is no vocabulary overlap, and also, the script used in both languages is different (Kanojia et al., 2020). Therefore, for developing an English-Marathi APE system, we use two separate encoders to encode *src* and *mt* (Junczys-Dowmunt and Grundkiewicz, 2018).

We apply transfer learning by using IndicBERT to initialize weights of the *src encoder* and the *mt encoder*. We choose IndicBERT as it is trained over text in Indian languages and English. We use a single transformer-based decoder that attends to representations of both *src* and *mt* and generates a post-edited version of the *mt*. We add one more cross-attention layer above the available cross-attention layer in the decoder. We pass the representation generated by *mt encoder* to the first cross-attention layer. The newly-added cross-attention layer receives two inputs: output of the first attention layer and representation generated by the *mt encoder*. Such placement allows the decoder to first attend to *mt*, which is prone to mistakes, and then it attends to *src*, which doesn't involve any errors. We share parameters between encoders, but the encoders generate different activations, and different attention layers receive the outputs of these encoders in the decoder. During the fine-tuning phase, we concatenate *mt* and *external MT candidate* using a special token '[SEP]' and pass this concatenated sequence to the *mt encoder*.

2.2 Sentence-Level Quality Estimation

In the Sentence-level Quality Estimation (QE) task, the machine-translated sentence is evaluated by human annotators by providing each instance with a Direct Assessment (DA) score (ranging from 0 to 100). These scores are then normalized using *z-score normalization*. A source sentence and the corresponding machine-translated output are passed to the sentence-level QE (sentence-QE) system as inputs, and it predicts a z-standardized DA score denoting the quality of translation.

We use the MonoTransquest (Ranasinghe et al., 2020), a XLM-R (Conneau et al., 2020) based model to obtain representations of the inputs. The XLM-R model is trained using a 2.5TB multilingual dataset retrieved from the CommonCrawl databases, which includes 104 languages. It is trained using the RoBERTa's masked language modelling (MLM) objective (Liu et al., 2019). We use the training (18K samples), and development (1K samples) sets shared in the WMT-22 Sentence-QE English-Marathi sub-task to train our sentence-QE model.

We use this sentence-QE model to rate the original translation and the output generated by our system. We then compare the ratings for both these sequences and select the one with a higher rating as the final output.

2.3 Curriculum Training Strategy (CTS)

We follow the CTS (Oh et al., 2021) to train our APE model. It involves gradually adapting a model to more complex tasks. In the first step, we train an encoder-decoder model for performing English to Marathi translation. We then add another encoder to the encoder-decoder model and train the resulting *dual-encoder single-decoder* model for the APE task using synthetic APE data in two phases. In the first phase, we train the APE model using APE triplets belonging to any domain except the General, News, and Healthcare domains. In the second phase, we train the model using synthetic APE triplets of the General, News, and Healthcare domains. Finally, we fine-tune the APE model using in-domain real APE data and external MT candidates.

2.4 Data Augmentation

Before using the synthetic APE data during the training steps of the CTS, we filter the low-quality triplets by using the LaBSE-based filtering (Feng et al., 2022). We do this to ensure adequate quality of the synthetic APE data. To do so, we first generate embeddings of the *src* and *pe* using the LaBSE model and normalize them. Then, we compute the cosine similarity between these normalized embeddings. If the cosine similarity is less than 0.91, we discard the corresponding APE triplet. Our experimental results show the importance of using good-quality APE data to train APE systems.

We also generate the phrase-level APE triplets using the good-quality synthetic APE data and the real APE data. We follow the procedure described by Sen et al. (2021) and extend it for the phrase-level triplet injection for APE. First, we use the Moses (Koehn et al., 2007) SMT system and train *src-mt* and *src-pe* phrase-based SMT systems. We then extract these phrase pairs from both SMT systems. In the next step, we collect pairs of phrase-pairs having same *src* from the *src-mt* and *src-pe* phrase tables. Finally, we follow the steps used in the LaBSE-based filtering and get cosine similarity scores for both the phrase pairs having the same *src*. If both the scores are more than 0.91, we combine these two phrase pairs to form a triplet and add it to the APE dataset.

To generate the external MT candidates, we train an mT5 (Xue et al., 2021) based English-Marathi NMT model over a publicly available English-Marathi parallel corpora (Samanantar (Ramesh et al., 2022), Anuvaad[1], Tatoeba[2], and ILCI (Bansal et al., 2013)) of around 6M parallel sentence pairs. We use the external MT candidates during the fine-tuning phase.

[1] Anuvaad: Github Repo

[2] Tatoeba Project

System	TER↓	BLEU↑
Do Nothing (Baseline)	22.93	64.51
+ CTS-based Training and External MT	20.08	67.39
+ LaBSE-based Data Filtering and in-domain training data	19.73	67.86
+ Phrase-level APE triplets	19.39	68.35
+ Sentence-level QE	**19.01**	**68.87**

Table 1: Results on the WMT-22 APE Development Set.

System	TER↓	BLEU↑
Do Nothing (Baseline)	20.28	67.55
IIT Bombay's Submission	**16.79**	**72.92**

Table 2: Results on the WMT-22 APE Test Set.

3 Experimental Setup

3.1 Dataset

This year's APE shared task focused only on the English-Marathi language pair. The real APE training data contains 18K APE triplets, and this APE data belongs to the General, Healthcare, and Tourism domains. The organizers also shared the synthetic APE data of various domains totaling around 25M APE triplets. As participants, we were permitted to use external data for this task.

To train a translation model, we use the publicly available English-Marathi parallel corpora of size around 6M parallel sentence pairs. For data augmentation, we first generate phrase-level APE triplets using synthetic and real APE data and then randomly select 50000 phrase-level APE pairs for augmenting with the synthetic APE data and 10000 for augmenting with real APE data.

3.2 Training Hyperparameters

We used NVIDIA DGX A100 GPUs for our experiments. We trained our models with a batch size of 32. We set the number of maximum epochs to 1000 with early stopping patience of 5. We used the Adam optimizer with a learning rate of 5 x 10^{-5}, $\beta_1 = 0.9$, and $\beta_2 = 0.997$. We set the number of warmup steps to 25K. On the decoder side, We used beam search with the beam size set to 5. For the LaBSE-based filtering, we used a threshold value of 0.91 for cosine similarity to ensure that *mt* and textitpe are similar to each other.

4 Results

In Table 2, we report the results of our APE system by evaluating it on the development set. To estimate the quality of our APE system output compared to the human-generated references, we use BLEU and TER score between the APE output and *pe*. Table 2 compiles the results of our experiments performed on the development set.

We compare the results of our experiments against a 'Do Nothing' APE baseline that simply outputs *mt* without any modification. When we trained our model using CTS and external MT candidates to increase feature diversity, the TER and BLEU scores improved to 20.08 TER points and 67.39 BLEU points from the baseline TER and BLEU scores of 22.93 and 64.51, respectively. The third row in the 2 shows the results of an experiment where we use a good-quality synthetic dataset for APE training obtained by filtering low-quality triplets using LaBSE-based filtering. The experiment also involves training the APE model in two phases: first, the model is trained on out-of-domain synthetic data and then on in-domain synthetic data. This setting brings -3.2 and +3.35 TER and BLEU score improvements over the baseline, and *underlines the importance of using good-quality in-domain APE data.*

The only change we make for performing the next experiment is augmenting the synthetic and real APE data using phrase-level APE triplets. Results of this experiment show that performance improves over the baseline by -3.54 TER points and +3.84 BLEU points. Towards the end, we also used a sentence-QE system to rate the original translation and the APE output. We then select one of them with a higher rating as the final output of our APE system. With the combination of the APE model and sentence-QE system, we see that the TER score improves to 19.01 points, and BLEU score increases to 68.87 points; which *shows that using the sentence-level QE system is an effective* *approach to discard APE output, in cases of over-correction.*

As per the information received by the shared task organizers, our APE system achieves a TER score of 16.79 points and a BLEU score of 72.92 when evaluated on the official test set, which is -3.49 TER points and +5.37 BLEU points improvement over the baseline.

5 Conclusions and Future Work

This paper presents our APE system submitted to the WMT-22 APE English-Marathi Shared task. We use a dual-encoder single-decoder model where both encoders are initialized using IndicBERT. We propose a new way to generate artificial phrase-level APE triplets by extending the phrase-pair injection method used in MT for APE. We show that augmenting APE training data with these phrase-level triplets and training the model with the CTS on good-quality in-domain APE data improves the performance of the APE system. Furthermore, we also explore using the sentence-level QE system to discard low-quality APE outputs. Evaluation of our APE system shows that our approach achieves significant gains on the WMT-22 APE development and test sets.

In future, we would like to extend this approach for automatic post-editing with the help of word-level quality estimation and come up with a single architecture for performing both the QE tasks along with APE. We would also like to attempt a multilingual APE system with a shared decoder across multiple languages.

6 Limitations

We use in-domain data to train the APE model in the last training stage and the fine-tuning stage. It makes the APE system robust in post-editing in-domain translations, but it also makes it sophisticated. We observe that the system's performance worsens when we pass out-of-domain translations to the system. Similarly, we observe poor performance when translations with distributional differences from the real APE data are passed to the APE system. We use a sentence-level QE system to compare the quality of the APE output and the original translation. Even though it helps us to get rid of poor-quality APE outputs, the APE system itself does not get benefited from it.

Acknowledgements

We would like to thank the anonymous reviewers. Their insightful comments helped us in improving the current version of the paper.

References

Akanksha Bansal, Esha Banerjee, and Girish Nath Jha. 2013. Corpora creation for indian language technologies–the ilci project. In *the sixth Proceedings of Language Technology Conference (LTC '13)*.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Rajen Chatterjee, Markus Freitag, Matteo Negri, and Marco Turchi. 2020. Findings of the WMT 2020 shared task on automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 646–659, Online. Association for Computational Linguistics.

Rajen Chatterjee, Matteo Negri, Raphael Rubino, and Marco Turchi. 2018. Findings of the WMT 2018 shared task on automatic post-editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 710–725, Belgium, Brussels. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 751–758, Berlin, Germany. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2018. MS-UEdin submission to the WMT2018 APE shared task: Dual-source transformer for automatic post-editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 822–826, Belgium, Brussels. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of EMNLP*.

Diptesh Kanojia, Raj Dabre, Shubham Dewangan, Pushpak Bhattacharyya, Gholamreza Haffari, and Malhar Kulkarni. 2020. Harnessing cross-lingual features to improve cognate detection for low-resource languages. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1384–1395, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Diptesh Kanojia, Prashant Sharma, Sayali Ghodekar, Pushpak Bhattacharyya, Gholamreza Haffari, and Malhar Kulkarni. 2021. Cognition-aware cognate detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3281–3292, Online. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Samuel Läubli, Mark Fishel, Gary Massey, Maureen Ehrensberger-Dow, and Martin Volk. 2013. Assessing post-editing efficiency in a realistic translation environment. In *Proceedings of the 2nd Workshop on Post-editing Technology and Practice*, Nice, France.

Jihyung Lee, WonKee Lee, Jaehun Shin, Baikjin Jung, Young-Kil Kim, and Jong-Hyeok Lee. 2020a. POSTECH-ETRI's submission to the WMT2020 APE shared task: Automatic post-editing with cross-lingual language model. In *Proceedings of the Fifth Conference on Machine Translation*, pages 777–782, Online. Association for Computational Linguistics.

WonKee Lee, Jaehun Shin, Baikjin Jung, Jihyung Lee, and Jong-Hyeok Lee. 2020b. Noising scheme for data augmentation in automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 783–788, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

António V. Lopes, M. Amin Farajian, Gonçalo M. Correia, Jonay Trénous, and André F. T. Martins. 2019. Unbabel's submission to the WMT2019 APE shared task: BERT-based encoder-decoder for automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 118–123, Florence, Italy. Association for Computational Linguistics.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. ESCAPE: a large-scale synthetic corpus for automatic post-editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Shinhyeok Oh, Sion Jang, Hu Xu, Shounan An, and Insoo Oh. 2021. Netmarble AI center's WMT21 automatic post-editing shared task submission. In *Proceedings of the Sixth Conference on Machine Translation*, pages 307–314, Online. Association for Computational Linguistics.

Santanu Pal, Sudip Kumar Naskar, and Josef van Genabith. 2016. Multi-engine and multi-alignment based automatic post-editing and its impact on translation productivity. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2559–2570, Osaka, Japan. The COLING 2016 Organizing Committee.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2022. Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. TransQuest at WMT2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1049–1055, Online. Association for Computational Linguistics.

Sukanta Sen, Mohammed Hasanuzzaman, Asif Ekbal, Pushpak Bhattacharyya, and Andy Way. 2021. Neural machine translation of low-resource languages using smt phrase pair injection. *Natural Language Engineering*, 27(3):271–292.

Abhishek Sharma, Prabhakar Gupta, and Anil Nelakanti. 2021. Adapting neural machine translation for automatic post-editing. In *Proceedings of the Sixth Conference on Machine Translation*, pages 315–319, Online. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jiayi Wang, Ke Wang, Kai Fan, Yuqi Zhang, Jun Lu, Xin Ge, Yangbin Shi, and Yu Zhao. 2020. Alibaba's submission for the WMT 2020 APE shared task: Improving automatic post-editing with pre-trained conditional cross-lingual BERT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 789–796, Online. Association for Computational Linguistics.

Daimeng Wei, Hengchao Shang, Zhanglin Wu, Zhengzhe Yu, Liangyou Li, Jiaxin Guo, Minghan Wang, Hao Yang, Lizhi Lei, Ying Qin, and Shiliang Sun. 2020. HW-TSC's participation in the WMT 2020 news translation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 293–299, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

LUL's WMT22 Automatic Post-Editing Shared Task Submission

Xiaoying Huang[1,2] and **Xingrui Lou**[1,2] and **Fan Zhang**[1] and **Mei Tu**[1]
[1]Samsung Research China – Beijing (SRC-B)
[2]State Key Laboratory of Media Convergence and Communication, Communication University of China
{huenhoying, louxingrui}@cuc.edu.cn, {zhang.fan, mei.tu}@samsung.com

Abstract

By learning the human post-edits, the automatic post-editing (APE) models are often used to modify the output of the machine translation (MT) system to make it as close as possible to human translation. We introduce the system used in our submission of WMT'22 Automatic Post-Editing (APE) English-Marathi (En-Mr) shared task. In this task, we first train the MT system of En-Mr to generate additional machine-translation sentences. Then we use the additional triple to bulid our APE model and use APE dataset to further fine-tuning. Inspired by the mixture of experts (MoE), we use GMM algorithm to roughly divide the text of APE dataset into three categories. After that, the experts are added to the APE model and different domain data are sent to different experts. Finally, we ensemble the models to get better performance. Our APE system significantly improves the translations of provided MT results by -2.848 and +3.74 on the development dataset in terms of TER and BLEU, respectively. Finally, the TER and BLEU scores are improved by -1.22 and +2.41 respectively on the blind test set.[1]

1 Introduction

Automatic Post-Editing (APE) is the task of automatically editing the translations of MT system. By using APE models, we can transfer MT system from general domain to specific domain and then reduce the workload of human post-edits. WMT has been holding APE task competitions in different languages and fields since 2015. Now the APE models are often based on transformer and improved on this basis.

WMT 2022's Automatic Post-Editing task focused on English-Marathi language pairs. The difference from the previous competition is that the target language is changed to Marathi. Besides, two new fields, medical care and tourism, are added. Participants are provided a training set with 18000 instances, a development set and a test set with 1000 instances respectively. Each dataset consists of source, machine-translation and post-edit triplets. The source sentences in English come from the healthcare, tourism, and general/news domains. The MT outputs are automatic translations to Marathi. The post-edits are human revisions of the target elements. This synthetic training data is prepared as a part of the 2022 APE shared task. The data is created by taking a parallel corpus, where the source data is translated using an MT system, and the references are considered as post-edits. Participants are also allowed to use any additional data for systems training.

Last year's research mainly focused on transfer learning and data augmentation. Sharma et al. (2021) utilizes the most advanced En-De machine translation model and further fine tune the APE dataset on this basis. We adopted the same strategy to train our baseline model with transfer learning and data augmentation. Due to the lack of a ready-made machine translation model of En-Mr as the basis of the APE model, we trained an APE model by using synthetic data and additional data. The APE model is then further fine-tuned with the APE dataset, which is data enhanced. In order to make use of the domain information in the training dataset, we use the mixture of experts structure and add adapter modules in the transformer, so that different adapters can learn the distribution of different domain information, thus improving the translation performance. The contributions of this work are as follows. (1) Data augmentation. We trained an external MT to obtain more data sets consistent with ape tasks. At the same time, we use Google translation to back translate the post-edits in the training set. The dataset is composed as follows: back translation <s> machine translation as input and post-edits as reference output. On the other hand, we take source <s> post-edits as

[1] Work performed during internship in Samsung Research China - Beijing

input and post-edits as reference output. (2) Mixture of adaptors. We implement the mixture of experts structure to deal with inputs from different domains, in which we use lightweight adapters as experts and introduce a classifier for expert routing. Considering the effect of directly initializing the adapter for training is not good enough, so we first set an adapter in the model and pre-train the model to obtain the adapter weight W_0. Since the training set comes from three different fields, we add two additional adapters, which will read W_0 as a parameter for initialization. We freeze all model parameters when training the model, and only fine tune the weights of the three adapters.

2 Related Work

Last year's WMT'21 APE shared task proved that both transfer learning and data augmentation were very effective. Facebook Fair's WMT19 news translation model was used in Shinhyeok's system (Oh et al., 2021). By continuously adding different levels of datasets, the model gradually understood APE tasks. For further improvement, Oh et al. (2021) used a multi-task learning strategy with dynamic weight average. By adding related subtasks, the model can learn unified representation. In addition, they also used the data set provided by ape shared task in previous years. Finally, their TER and BLEU scores were 17.28 and 71.55, respectively. Oh et al. (2021) used the most advanced machine translation model as the pre-trained model. The WikiMatrix dataset was uesd to make the model distribution tend to match the field. After that, APE samples from former years were added for fine adjustment. Finally, their model's TER and BLEU scores were 17.85 and 70.5, respectively.

Considering the experience of previous competitions, we used the existing data to train an En-Mr translation model as a data augmentation method due to the lack of advanced En-Mr translation model. Inspired by the MoE, the built-in adapter module enables the model to learn three data distributions at the same time to improve the performance of translation.

3 Dataset

3.1 Data Source

We used the WMT22 official English-Marathi APE dataset which consisted of a training and development set. We also used synthetic training data, which was prepared as a part of the 2022 APE shared task. In addition, we collected LoResMT2021 Shared Task (mac) data, CVIT PIBv1.3 (Philip et al., 2021), bible-uedin (Christodouloupoulos and Steedman, 2015) as some additional data to train our models. The LoResMT2021 Shared Task focused on machine translation of COVID-19 data for both low-resource spoken and sign languages. The LoResMT2021 dataset contains three parts: English-Irish, English-Marathi, and Taiwanese Sign language-Traditional Chinese. We only use its English-Marathi parallel corpora. CVIT PIBv1.3 is used in this work as a source for articles published in several Indian Languages to extract a multiparallel corpus. Sentences in CVIT PIBv1.3 aligned parallel corpus between 11 Indian languages, crawling and extracting from the press information bureau website. Bible-uedin is a multilingual parallel corpus created from translations of the Bible compiled by Christos Christodoulopoulos and Mark Steedman. The summary of the corpora used is provided in Table 1.

3.2 Data augmentation

As shown in Table 1, we have collected lots of parallel corpus but these corpus lack the MT part (LoResMT2021, CVIT PIBv1.3, Bible-uedin, and some synthetic training data). Following the method of generating synthetic training data, we first train a machine translation system, and then use this system to translate the source data. To generate translation similar to synthetic training data as much as possible, we did not use *src-pe* pairs but *src-mt* pairs when training MT models. We use all parallel corpus to train MT model. We are able to achieve a BLEU score of 25.3 with our MT model. Finally, we translate the sources we collected by our MT model and achieve approximately 2500000 triplets.

Yang et al. (2020) utilized data augmentation with external MT to generate the external translated sentence, which could help generate the post-editing sentence. We take a similar line of approach by leveraging external MT to generate the external translated sentence but the result is not satisfactory. So we utilize external MT to generate the external back-translations. We'd like to use back-translations to add a set of parallel corpora for the model to learn the rules of post-edits. In addition,

Source	pairs	type
APE dataset	18k	*src-mt-pe*
Synthetic training data	2.57m	*src-mt-pe*
LoResMT2021	21k	*src-pe*
CVIT PIBv1.3	117k	*src-pe*
Bible-uedin	60k	*src-pe*

Table 1: Publicly available corpuses for Indian languages.

we also use sentence X that contains a source sentence (*src*) and a post-editing sentence (*pe*) as input. We assume that the model can learn the invariance in post-editing rules by leaking some information of *pe*.

In this paper, we use D_{ape} for [*src*, <*s*>, *mt*], D_{bac} for [*src'*, <*s*>, *mt*] and D_{pe} for [*src* and <*s*>, *pe*].

4 Model

We describe our baseline model followed by the details of domain and task adaptation in this section.

4.1 Fine-tuned Transformer

Compared with previous APE tasks, this task focuses on English-Marathi language pairs. It is impossible to fine tune APE dataset on the basis of MT model. We decided to solve the APE task as NMT alike task. To adapt this idea with Transformer, we use a special token <*s*> to concatenate *src* and *mt* to generate input sentence: [*src*, <*s*>, *mt*]. We first trained the APE model with the standard Transformer (Vaswani et al., 2017) structure using synthetic training data and additional data. In order to fix the mismatch between the APE model training data and the distribution in our task, we further fine-tuned the APE model on the APE dataset.

To further solve the problem of limited data, we use the data collected in the Data section to adopt three data augmentation methods. First, we use Google translation system to create the *src'* from the provided *pe* text. We simply concatenate the *src'* with *mt* to form the new input: [*src'*, <*s*>, *mt*]. After this, the model input consists of [*src*, <*s*>, *mt*] and [*src'*, <*s*>, *mt*], which contains 36000 triplets. The second method is to add [*src*, <*s*>, *pe*] as the input on the basis of the original input and the third method is to add the first two as input at the same time. In the first way, we'd like to add a group of parallel corpora for the rules in editing after model learning. The second way is to think that by adding PE, the model can learn the rules of human post editing from SRC, MT and PE. The purpose to adopt the third method is to combine the first two methods for a better model performance.

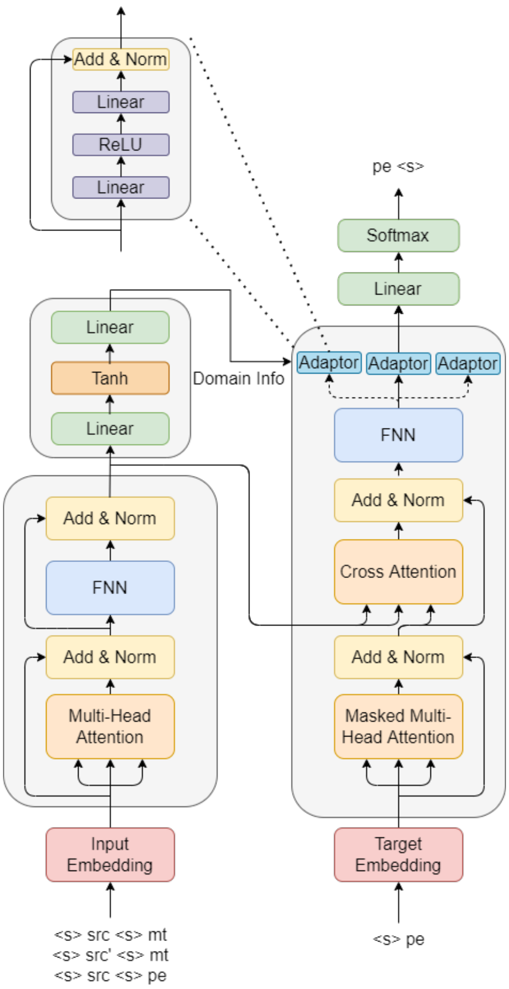


Figure 1: Adapter overall framework.

4.2 Adapter

We found that the APE dataset contains medical, tourism and general/news data. Inspired by the mixture of experts (Jacobs et al., 1991), we introduce adapters (Bapna and Firat, 2019; Pham et al., 2020) to handle different domains. We suppose that different adapters can process different domain data, so as to keep other parameters unchanged to improve the translation performance of the model for each domain, thus improving the overall TER and BLEU.

The structural diagram of the adapter is shown in Figure 1, which is similar to the FFN layer in transformer, but has a low dimensional hidden layer for nonlinear activation. In the experiment, we add the adapter layer after the FFN layer for each block in the decoder. Each adapter layer consists of three adapters. A classifier is introduced after the encoder to generate domain information (Domain Info) and it decides which adapter is activated during inference. Overall, in the inference phase, the classifier first generates domain information (Domain Info) by the encoder output, and then the corresponding adapter in each decoder is activated by the domain information. Finally, same with a general NMT model, the output is generated with an auto-regressive process.

The adapter model is trained with a pipe-line training process. First, an APE model is trained as the base model. Different from the original baseline model, an adapter is injected to each decoder layer and will be used to initialize the other two adapters in the same decoder layer. Then the classifier is trained by using a multi-classification task. Finally, with all other parameters are frozen, the parameters of the adapters are optimized by using the NMT task.

5 Experiment and Results

5.1 Experimental Settings

Both our En-Mr MT model and APE model are implemented with Fairseq framework (Ott et al., 2019). The Transformer model used for both models is Transformer-base with 6 encoders and 6 decoders, and the hidden size is 2048 for FFN layers and 512 for all other layers. The adapter used in our model is also modified to have a larger parameter size, where the hidden size of the inner layer is set to 2048.

Because we lacked the MT model, we learned the vocabulary of En and Mr by BPE. Specially, for English, we use token first and then BPE, while for Marathi, we directly conduct BPE. We believe that if token is used for Marathi before BPE, the model cannot learn the rules for punctuation after manual post-edits. The thing we should notice that the vocabulary of the En-Mr model cannot be shared which contains 31K and 31K sub-tokens for En and Mr respectively. Since the input of APE model contains En and Mr, the joint vocabulary of APE should be the total number of tokens in both languages, about 58K sub-tokens. All models were trained on NVIDIA Tesla V100. We use Adam optimizer to optimize with a fixed learning rate of 5e-4. The max tokens are set to 4096, about 64 batch sizes.

System	BLEU	TER
baseline	64.62	19.93
+Fine-tuning (D_{ape})	66.19	18.71
+Fine-tuning (D_{ape}+D_{bac})	66.44	18.56
AVG_FT (D_{ape}+D_{bac})	66.94	18.06
+Fine-tuning (D_{ape}+D_{pe})	67.24	18.09
AVG_FT (D_{ape}+D_{pe})	**67.37**	**17.91**
+Fine-tuning (D_{ape}+D_{bac}+D_{pe})	66.93	18.30
AVG_FT (D_{ape}+D_{bac}+D_{pe})	67.22	17.93

Table 2: This is the experimental result of fine-tune. AVG represents the weighted average of the model.

System	BLEU	TER
baseline	64.62	19.93
Adpt (D_{ape}+D_{bac})	66.89	18.34
AVG_Adpt (D_{ape}+D_{bac})	66.84	18.36
Adpt (D_{ape}+D_{pe})	67.55	17.90
AVG_Adpt (D_{ape}+D_{pe})	67.55	17.89
Adpt (D_{ape}+D_{bac}+D_{pe})	**67.71**	**17.85**
AVG_Adpt (D_{ape}+D_{bac}+D_{pe})	67.67	17.89

Table 3: This is the experimental result of adapter. AVG represents the weighted average of the model.

5.2 Fine-tuned Transformer

Table 2 shows the experimental results of APE finetune model, where the baseline result is produced by directly calculating scores between the provided *mt* and *pe*. The first experiment is performed by fine-tuning all parameters of the pre-trained Transformer on the official training set. The TER and BLEU on the 2022 dev set were 18.71 and 66.91, which were -1.2 and + 1.57 better than baseline. This demonstrates that fine-tuning the pre-trained NMT model on the limited dataset can be useful.

The experiment of training model on D_{ape}+D_{bac} and D_{ape}+D_{pe} for data augmentation shows significant improvements on the performance. However, after performing experiments with different checkpoints of APE model, we find that the best checkpoint is not the best saved checkpoint for translation, which motivates us to average model weight parameters near the best checkpoint. The avg model results show averaging the model weight parameters near the best checkpoint can help the model to be closer to the convergence point locally. The performance of the model is improved well.

5.3 Adapter

Table 3 shows the experimental results of APE adapter model. For D_{ape}+D_{bac} dataset, adding the

System	BLEU	TER
baseline	67.55	20.28
Finetune_PRIMARY	69.66	19.36
Adapter_CONTRASTIVE	69.96	19.06

Table 4: Results on test dataset.Finetune_PRIMARY ensembles AVG_FT (D_{ape}+D_{bac}) and AVG_FT (D_{ape}+D_{pe}). Adapter_CONTRASTIVE ensembles AVG_Adpt (D_{ape}+D_{bac}) and AVG_Adpt (D_{ape}+D_{pe}).

adapter does not improve the APE performance of the model, but for D_{ape}+D_{pe} dataset, adding adapter makes the model reduce TER and improve BLEU, reaching the lowest TER and the highest BLEU respectively. The experimental results show that the rough classification of data and the learning of their respective distributions are more conducive to the better APE performance of the model.

5.4 Results on Test set

Table 4 shows the official results of our proposed methods on WMT22 test dataset with a baseline scores of 20.28 and 67.55, which is higher than the development dataset with 19.93 and 64.62 in terms of TER and BLEU. Despite its high quality, our proposed methods show effectiveness on this test dataset. We find that there are some Arabic numerals and Devanagari numerals in post-edits. However, because we are not familiar with Marathi, we do not know the number modification rules. Therefore, we replace all Arabic numerals in test results with Devanagari numerals to get the final post-edits.

6 Conclusion

In this paper, we first use the data augmentation method to build the *src' <s> mt* and *src <s> pe* as two additional training datasets. We suppose that the enhanced datasets can effectively improve the performance of the APE model. The experimental results show that the data augmentation method we used is effective. At the same time, it also shows that adding *pe* can make the model automatically learn the rules of human post-edits. After that, we draw lessons from mixture of experts. We add adapters in the APE baseline model. And we let the training data be sent to different adapters through the trained classifier so that the model can further learn the post-editing rules in different translations. The experimental results confirm that our system can modify the output of MT system with high efficiency and quality. Compared with baseline, the TER and BLEU scores are improved by -1.22 and + 2.41 respectively.

References

machinetranslate.org. https://machinetranslate.org/. Accessed: 2022-05-12.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548.

Christos Christodouloupoulos and Mark Steedman. 2015. A massively parallel corpus: the bible in 100 languages. *Language resources and evaluation*, 49(2):375–395.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Shinhyeok Oh, Sion Jang, Hu Xu, Shounan An, and Insoo Oh. 2021. Netmarble ai center's wmt21 automatic post-editing shared task submission. In *Proceedings of the Sixth Conference on Machine Translation*, pages 307–314.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Minh Quang Pham, Josep-Maria Crego, François Yvon, and Jean Senellart. 2020. A study of residual adapters for multi-domain neural machine translation. In *Conference on Machine Translation*.

Jerin Philip, Shashank Siripragada, Vinay P Namboodiri, and CV Jawahar. 2021. Revisiting low resource status of indian languages in machine translation. In *8th ACM IKDD CODS and 26th COMAD*, pages 178–187.

Abhishek Sharma, Prabhakar Gupta, and Anil Nelakanti. 2021. Adapting neural machine translation for automatic post-editing. In *Proceedings of the Sixth Conference on Machine Translation*, pages 315–319.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Hao Yang, Minghan Wang, Daimeng Wei, Hengchao Shang, Jiaxin Guo, Zongyao Li, Lizhi Lei, Ying Qin, Shimin Tao, Shiliang Sun, et al. 2020. Hw-tsc's participation at wmt 2020 automatic post editing shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 797–802.

Findings of the WMT 2022 Biomedical Translation Shared Task: Monolingual Clinical Case Reports

Mariana Neves[1] *Antonio Jimeno Yepes[2] Amy Siu[3]
Roland Roller[4] Philippe Thomas[4] Maika Vicente Navarro[5]
Lana Yeganova[6] Dina Wiemann[7] Giorgio Maria Di Nunzio[8]
Federica Vezzani[8] Christel Gerardin[9] Rachel Bawden[10]
Darryl Johan Estrada[11] Salvador Lima-López[11] Eulàlia Farré-Maduell[11]
Martin Krallinger[11] Cristian Grozea[12] Aurélie Névéol[13]
[1]German Centre for the Protection of Laboratory Animals (Bf3R),
German Federal Institute for Risk Assessment (BfR), Berlin, Germany
[2]RMIT University, Australia
[3]Berliner Hochschule für Technik, Germany
[4]German Research Center for Artificial Intelligence (DFKI), Berlin, Germany
[5]Leica Biosystems, Australia
[6]NCBI/NLM/NIH, Bethesda, USA
[7]Novartis AG, Basel, Switzerland
[8]Dept. of Linguistic and Literary Studies University of Padua, Italy
[9]Sorbonne Université, Inserm, IPLESP, Paris, France
[10]Inria, Paris, France
[11]Barcelona Supercomputing Center, Spain
[12]Fraunhofer Institute FOKUS, Berlin, Germany
[13]Université Paris-Saclay, CNRS, LISN, Orsay, France

Abstract

In the seventh edition of the WMT Biomedical Task, we addressed a total of seven language pairs, namely English/German, English/French, English/Spanish, English/Portuguese, English/Chinese, English/Russian, English/Italian. This year's test sets covered three types of biomedical text genre. In addition to scientific abstracts and terminology items used in previous editions, we released test sets of clinical cases. The evaluation of clinical cases translations were given special attention by involving clinicians in the preparation of reference translations and manual evaluation. For the main MEDLINE test sets, we received a total of 609 submissions from 37 teams. For the ClinSpEn sub-task, we had the participation of five teams.

1 Introduction

This is the seventh edition of the biomedical translation task offered under the umbrella of the Conference on Machine Translation (WMT22).[1] This shared task builds on the six previous editions of the biomedical translation task (Bojar et al., 2016; Jimeno Yepes et al., 2017; Neves et al., 2018; Bawden et al., 2019, 2020; Yeganova et al., 2021). Similar to previous years, we addressed seven language pairs, in both directions, namely: German/English (de2en and en2de), Spanish/English (es2en and en2es), French/English (fr2en and en2fr), Italian/English (it2en and en2it), Portuguese/English (pt2en and en2pt), Russian/English (ru2en and en2ru), and Chinese/English (zh2en and en2zh).

In the biomedical translation task this year, participants were asked to translate shared test sets (described in Section 2) comprising documents belonging to three different text genres: scientific abstracts, clinical cases and terminology items. In total, seven language pairs (14 translation directions) were included this year, with both low-resource and high-resource pairs. For each language direction, we provide baseline systems relying on pre-trained

[*] The contribution of the authors are the following: MN prepared the MEDLINE test sets, performed test set validation, manual validation, and organized the task; AJY performed test sets validation, manual validation and the automatic evaluation; RR, PT, MVN, LY, DW. GMDN, FV performed test sets validation and manual validation; CGn created reference translation and performed manual validation; RB performed manual validation; DJE, SLL, EFM, MK organized the ClinSpEn sub-task; CGa created the baselines; and AN collected information on participants' methods, performed test sets validation, manual validation and created reference translation. All authors approved the final version of the manuscript. E-mail for contact: mariana.lara-neves@bfr.bund.de

[1] https://www.statmt.org/wmt22/biomedical-translation-task.html

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 694 - 723
December 7-8, 2022 ©2022 Association for Computational Linguistics

neural translation models (described in Section 3). In order to help gain insight into the system performance, we collected information on the specific material and methods used in the systems from the participants (Section 4). System outputs for each task were evaluated both automatically and manually (as described in Section 5 and 6, respectively). One particular growth direction that we explored this year was the inclusion of full clinical case descriptions. A small set of five clinical cases in English were included in the MEDLINE test sets from English and a larger clinical corpus corpus was also included in the ClinSpEn track. We also involved clinicians in the preparation of gold standard translations and manual evaluation of clinical cases for en2fr and en2es.

Two types of submissions were received for the MEDLINE test sets: those submitted using (i) our submission system (hereafter called BioWMT), as in previous years, and (ii) the OCELoT submission system,[2] which was also used in the WMT general task.

In addition, an independent subtask was held as part of the Shared Task: ClinSpEn.[3] ClinSpEn focuses on the automatic translation of clinical content in both English and Spanish. Three subtracks are proposed based on different possible use cases: clinical case reports, clinical terminology obtained from literature and Electronic Health Records (EHR) and ontology concepts. Unlike the rest of the tasks, ClinSpEn's evaluation was done through CodaLab[4] and new submissions can still be made.

2 Test sets

In this section we describe the various test sets that we released for this year's edition of the WMT Biomedical task.

2.1 MEDLINE test sets

The MEDLINE test sets consisted of abstracts and case reports from the MEDLINE database. We aimed to retrieve 50 articles for each language direction. For the directions into English, the test sets consisted only of parallel abstracts. For the directions from English, we manually selected five clinical case reports, which were only available in English and which were the same across all language pairs. We completed each of these test sets with parallel abstracts. Table 1 summarizes the MEDLINE test sets as released in our submission system and in OCELoT. The only difference between the test sets released in our submission system and in OCELoT was that the latter contained aligned sentences, as provided by the automatic alignment. We describe the construction of the parallel abstracts and clinical case reports below.

2.1.1 Parallel abstracts

For the parallel abstracts, we downloaded the MEDLINE database[5] around the end of February and selected parallel abstracts for each language pair. We targeted publications whose PMID (PubMed identifier) was not included in any of our previous test sets and training data. We processed the abstracts using the same tools for sentence splitting and sentence alignment as in previous years (Yeganova et al., 2021). We manually checked the quality of the alignment using the Appraise tool (Federmann, 2010) and present results in Table 2.

2.1.2 Clinical case reports

For test sets *from English*, we decided to select clinical case presentations in order to include documents that would be closer in genre to clinical narratives found in patient records. Five clinical cases[6] were selected from publications of the *Journal of Medical Case Reports* (an open access publication) according to the following criteria:

- Reports a case related to oncology (based on the expertise of clinicians that agreed to contribute to the evaluation);
- Reports containing specific values such as lab results;
- Reports containing a limited amount of references to images and tables (to maximize resemblance with EHR narrative);

Both the abstract of the article and the full case presentation were included in the test set.

A gold standard translation of the clinical cases (both abstract and full case presentations) was created for French. We used the free version of

[2] https://github.com/AppraiseDev/OCELoT

[3] https://temu.bsc.es/clinspen

[4] https://codalab.lisn.upsaclay.fr/competitions/6696

[5] https://www.nlm.nih.gov/databases/download/pubmed_medline.html

[6] PMIDs: 19144122, 21838907, 35303936, 35313981, 35144678

Pairs	Documents			Sentences (WMTBio)		Sentences (OCELoT)	
	Mono.	Parallel	Total	Mono.	Parallel	Mono.	Parallel
de2en	-	50	50	-	434/453	-	419
en2de	5	45	50	210/-	462/467	210	435
es2en	-	50	50	-	459/461	-	436
en2es	5	45	50	210/-	397/404	210	377
fr2en	-	50	50	-	319/325	-	308
en2fr	5	45	50	210/-	608/609	210	590
it2en	-	43	43	-	457/461	-	427
en2it	5	39	44	210/-	372/364	210	327
pt2en	-	50	50	-	459/478	-	454
en2pt	5	45	50	210/-	465/454	210	443
ru2en	-	50	50	-	408/398	-	351
en2ru	5	45	50	210/-	526/545	210	453
zh2en	-	48	48	-	281/409	-	277
en2zh	5	45	50	210/-	424/362	210	359

Table 1: Number of documents and sentences in the MEDLINE test sets. For the Ocelot test sets, the test sets have the same number of sentences for both languages in a pair.

Language	OK	Source>Target	Target>Source	Overlap	No Align.	Total
de2en	358 (85.2%)	26 (6.2%)	14 (3.3%)	7 (1.7%)	15 (3.6%)	420
en2de	383 (87.0%)	28 (6.4%)	13 (3.0%)	4 (0.9%)	12 (2.7%)	440
es2en	367 (83.4%)	32 (7.3%)	11 (2.5%)	11 (2.5%)	19 (4.3%)	440
en2es	350 (90.9%)	11 (2.9%)	14 (3.6%)	2 (0.5%)	8 (2.1%)	385
fr2en	253 (84.7%)	21 (7.0%)	6 (2.0%)	1 (0.3%)	18 (6.0%)	299
fr2en §	288 (93.6%)	5 (1.6%)	5 (1.6%)	2 (0.6%)	8 (2.6%)	308
en2fr	450 (86.8%)	64 (12.4%)	1 (0.2%)	-	3 (0.6%)	518
en2fr §	590 (97.8%)	13 (2.2%)	-	-	-	603
it2en	340 (79.0%)	44 (10.2%)	19 (4.4%)	14 (3.2%)	14 (3.2%)	431
en2it	261 (75.9%)	21 (6.1%)	16 (4.6%)	4 (1.2%)	42 (12.2%)	344
pt2en	426 (93.8%)	17 (3.7%)	8 (1.8%)	3 (0.7%)	-	454
en2pt	365 (82.2%)	36 (8.2%)	14 (3.1%)	7 (1.6%)	22 (4.9%)	444
ru2en	226 (64.4%)	25 (7.1%)	17 (4.8%)	7 (2.0%)	76 (21.7%)	351
en2ru	281 (61.2%)	32 (7.0%)	30 (6.5%)	25 (5.5%)	91 (19.8%)	459
zh2en	264 (94.0%)	4 (1.4%)	8 (2.8%)	-	5 (1.8%)	281
en2zh	346 (95.9%)	3 (0.8%)	5 (1.4%)	-	7 (1.9%)	361

Table 2: Statistics (number of sentences and percentages) of the quality of the automatic alignment for the MEDLINE test sets. § Results after manual correction of sentence segmentation and/or alignment.

DeepL[7] followed by two rounds of post-edition: first, a native French speaker with formal translation training and knowledge of clinical text (AN) post-edited the machine translation (MT) focusing on linguistic quality and fluidity of the translation; second, a clinician (CG) post-edited the revised text focusing on clinical correctness and adequacy of the text with the French clinical narrative genre. In this second step, special attention was given to values such as lab results, which can be expressed using different units in English vs. French. The goal was to produce a translation that would convey properly processed information for direct use by a clinician. We computed BLEU scores between the original machine translated text and successive rounds of post-edition. BLEU between MT and the final gold standard translation was 38 for abstracts and 42 for full texts, while BLEU between the translator post-edited text and final gold standard translation was 63 for abstracts and 85 for full texts.

[7] http://www.DeepL.com/Translator

2.2 ClinSpEn test sets

For each of the ClinSpEn sub-tracks, a gold standard dataset was prepared with human translations created by domain experts. Additionally, a big collection of monolingual background data was provided for each subtrack so that participants could test the scalability of their systems or use them for other purposes.

Sub-track 1: Clinical case reports. This subtrack deals with the translation of clinical case reports. Clinical cases are a text genre where a patient's current condition, medical history, clinical presentation, examinations, treatment and diagnosis are described. They can be pretty similar to EHR both in form and content. However, unlike EHR, clinical cases are often free of privacy-related issues. This means that they can be used as substitute to train NLP systems for the clinical domain.

The gold standard dataset's clinical cases were carefully selected to cover a wide range of aspects related to COVID-19: different types of patients (children, adults, elderly and pregnant people, babies), different comorbidities (cancer, mental health issues, immunosuppressed patients) and symptomatology (mild and severe presentations, dermatologic, immunologic and psychiatric manifestations, thrombosis, etc.). The reports were translated from English to Spanish by a professional medical translator in a first step and revised by a clinical expert in a second step. The background set includes around 3,800 clinical case reports in English extracted from PubMed Central.

The dataset includes a total of 202 COVID-19 clinical case reports (50 for the dev set, 152 for the test set) and the direction of this sub-track is en2es.

Sub-track 2: Clinical terminology. This subtrack deals with the translation of clinical terminology. Translating clinical terminology is very relevant due to the existence of many established concepts and multi-word expressions (MWE) that need to be translated not only correctly but also consistently. Systems able to consider not only full sentences but also specific terms are able to provide more accurate translations, something fundamental in the clinical domain.

The gold standard terms were extracted from biomedical literature and electronic health records using information retrieval systems, filtered and translated and revised by professional medical translators. Amongst other semantic classes, the selected terms include diseases, symptoms and findings, procedures, drugs and species. The background set includes over 200,000 concepts in Spanish from the same sources.

The dataset includes a total of 19,128 terms (7,000 for the dev set and 12,128 for the test set). The direction of this sub-track is es2en.

Sub-track 3: Clinical terminology. This subtrack deals with the translation of concepts extracted from ontologies. Ontologies are one of the main ways of structuring knowledge. In the clinical domain, they are widely used mainly to normalize the content of electronic health records. However, their everyday use can be greatly limited by their unavailability in languages other than English. MT systems specifically trained for this type of data can be of great help to improve the impact of these ontologies or to ease a manual translation process.

The gold standard for this task is made up of concepts extracted from various free-access biomedical ontologies and taxonomies and then manually translated by a professional medical translator. Due to their origin, these concepts may present different challenges than terms extracted from free text, such as semi-structured concepts. The background set includes 300,000 concepts in English extracted from the same sources.

The dataset includes a total of 2,189 concepts (300 for the dev set and 1,789 for the test set). The direction of this sub-track is en2es.

3 Baselines

The baselines for en2de, en2fr, en2es, en2pt, de2en, fr2en, es2en, and pt2en were computed using models we trained ourselves in the previous years using Marian NMT (Junczys-Dowmunt et al., 2018). The baselines for en2zh, en2it, en2ru, zh2en, it2en, zh2en were computed using pre-trained Marian models distributed as HuggingFace "Transformers" library models,[8] without trying to increase their performance on the biomedical texts through further fine-tuning. The computation was performed on a single Nvidia A5000 GPU card.

The baselines are strongly outperformed by the participants of the biomedical task, with the exception of **en2it** where all reach similar and very high levels, in excess of 47 BLEU. Especially our zh2en baseline needs improvement.

[8] https://huggingface.co/Helsinki-NLP

4 Teams and systems

In this section we describe the teams and the number of submissions that we received from our two submission systems. When considering both the MEDLINE and the ClinSpEn sub-task, we had a total of 40 participating teams. We describe the submissions for each of them below.

4.1 MEDLINE participation

This year, we received a total of 609 submissions from 37 teams (see Table 3), from the following countries: China (7), France (2), Poland (1), Russia (1), and South Korea (1). Most teams (N=25), however, did not report a country of affiliation.

The number of submissions for each of the MEDLINE test sets are split into to parts: from English in Table 4 and into English in Table 5. We received around 100 more submissions for the test sets into English (354 vs. 255).

As in the 2020 and 2021 editions, we asked participants to fill out a survey with key information regarding the specific material and methods used in their self-identified primary runs used for manual evaluation. The survey comprised 15 questions covering the translation methods and corpora used. For consistency with previous years, the only change to the questionnaire was the addition of a question regarding the method used by teams to estimate the environmental impact of their experiments. We included the CO2 measurement methods identified in (Bannour et al., 2021) as options.

Only six teams supplied information about their "best run", and none reported measuring the environmental impact of their participation to the task. On average, the time spent by participants to supply information for one language pair was 7 minutes and 13 seconds (median: 3 minutes and 27 seconds). This is consistent with the previous survey statistics and suggests that the time commitment for supplying this information is limited, even for teams addressing more than one language pair.

All teams used transformer-based neural MT (NMT), relying mostly on existing implementations. Contrarily to last year, teams addressing several language pairs adapted their setup across them. See Table 6 for details of the teams' methods.

For in-domain data, teams used the training data distributed as part of the task as well as many of the sources described in (Névéol et al., 2018). Additional corpora used for Chinese were prepared by the teams but are not always available or described in detail, except for ParaMed, which relied on the New England Journal of Medicine to create a parallel corpus (Liu and Huang, 2021). Terminologies used by team Summer are available online.[9] The in-domain monolingual corpora used often use different selections of MEDLINE. We can also notice that the use or pre-processing of the same resources can differ between teams as the size reported for seemingly similar data can differ significantly. Table 7 provides details of the in-domain data used by the teams.

For relevant language pairs, parallel data from other WMT tracks (e.g. General or News Task) was used. Out-of-domain data was also used in the form of pre-trained base models. Table 8 shows details of the out-of-domain data used by the teams.

4.2 ClinSpEn Participation

In total, 11 different teams both from academia and industry registered for the ClinSpEn subtask, although only 5 teams ended up submitting their predictions. Four of them participated in all subtracks, with one of them participating only in subtrack 2 (clinical terminology translation). Table 9 presents an overview of the teams who submitted their predictions to the task.

5 Automatic evaluation

In this section we present the automatic evaluation that we performed for the MEDLINE and the ClinSpEn test sets.

5.1 MEDLINE test sets

For the MEDLINE test sets, we calculated the BLEU scores in the same way as previous years (Yeganova et al., 2021). We split the runs that we received into three groups: (i) runs to our BioWMT submission system; (ii) runs to the OCELoT Biomedical Task; and (iii) runs to the OCELoT General Task. As already discussed above, the only difference between the test sets in OCELoT and the ones in our submission system is that the sentences are aligned in OCELoT.

Results for runs to our BioWMT submission systems are presented in Tables 10 and 11. Runs for the Biomedical Task in OCELoT are shown in Tables 12 and 13. The run identifiers were mapped to names (e.g. run1, run2), and the mapping is presented in the Appendix (Tables 23 and 24). Finally,

[9] https://github.com/neulab/covid19-datashare/tree/master/parallel/terminologies

Team ID	Institution	Lime Survey	Publication
AISP-SJTU	AI Speech Co. and Shanghai Jiao Tong University, China		
ALMAnaCH-Inria	Inria, France		
aoligei	-		
bhcs-mt	-		
ChicHealth	ChicHealth, China		
DLUT	-		
DTrans	-		
DTranx	-		
ECNU-MT	East China Normal University, China	✓	(Zheng et al., 2022)
eTranslation	European Commission		
GTCOM	-		
Huawei-BabelTar	Huawei Technologies	✓	(Wang et al., 2022)
Huawei-TSC	Huawei Technologies	✓	(Wu et al., 2022)
JDExploreAcademy.Vega-MT	-		
KwaiMT	-		
Lan-BridgeMT	Lan-Bridge, China		
LanguageX	-		
LT22	-		
Manifold	-		
MeteorMan	-		
neunlplab	-		
njupt-mtt	-		
ONLINE-A	-		
ONLINE-B	-		
Online-G	-		
ONLINE-W	-		
ONLINE-Y	-		
OpenNMT	-		
PAHT	-		
PROMT	PROject MT, Russia		
SPECTRANS	Université Paris Cité, France	✓	(Ballier et al., 2022)
SRPOL	Samsung Research, Poland		
SRT	Samsung Research, South Korea	✓	(Choi et al., 2022)
Summer	Tencent, China	✓	(Li et al., 2022)
super_star	-		
szdx	-		
taicangshaxigaozhong	-		
ustc-mt	-		
V2ray	-		

Table 3: List of the participating teams.

due to the large number of teams and runs, we split the General Task runs into various results tables. The from-English submissions are split into two parts in Tables 14 and 15, while the identifier mapping is provided in Tables 25 and 26. Similarly, the into-English submissions are split into two parts in Table 16 and 17, while the identifier mapping is provided in Tables 27 and 28.

In general, the scores were much higher for runs to the BioWMT submission system than for the ones from the OCELoT test sets. All runs for the BioWMT submissions system outperformed our baseline. We did not provide a baseline for the OCELoT test sets.

5.2 ClinSpEn - CodaLab

The ClinSpEn subtask was evaluated in the CodaLab platform (Pavao et al., 2022). CodaLab is an open-source platform for running competitions, with some of its main advantages being automatic scoring and leaderboard building.

ClinSpEn submissions were evaluated using five common MT metrics: COMET (Rei et al., 2020), METEOR (Banerjee and Lavie, 2005), SacreBLEU (Post, 2018), BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004). The main metric used for comparison is SacreBLEU, which is the same as OCELoT uses, and the other metrics are given so that participants are able to evaluate their systems from different perspectives. Part of the evaluation scripts were shared by the MedMTEval organizers

Teams	en2de	en2es	en2fr	en2it	en2pt	en2ru	en2zh	Total
AISP-SJTU	-	-	-	-	-	-	G2	2
ALMAnaCH-Inria	-	-	-	-	-	G2	-	2
aoligei	-	-	-	-	-	-	O2G2	4
bhcs-mt	-	-	-	-	-	-	G4	4
ChicHealth	-	-	-	-	-	-	B1	1
DLUT	-	-	-	-	-	-	G4	4
Dtranx	O1G3	O2	O3	O3	O3	O2G2	O2G2	23
eTranslation	-	-	-	-	-	G3	-	3
ECNU-MT	-	-	-	-	-	-	B1	1
GTCOM	-	-	-	-	-	-	G3	3
Huawei-BabelTar	B3	B3	B3	B3	B3	B3	B3	21
Huawei-TSC	B3O6	-	B3O4	-	-	B3O3	B3O6G7	38
JDExploreAcademy.Vega-MT	G2	-	-	-	-	G2	G7	11
KwaiMT	-	-	-	-	-	-	G3	3
Lan-BridgeMT	O2G4	-	-	-	-	O2G4	O4G4	20
LanguageX	-	-	-	-	-	-	G4	4
Manifold	-	-	-	-	-	-	G7	7
MeteorMan	-	-	-	-	-	-	G1	1
neunlplab	-	-	-	-	-	-	G6	6
njupt-mtt	O1	-	O3	-	-	O3G4	O3G7	21
ONLINE-A	G1	-	-	-	-	G2	G1	4
ONLINE-B	G1	-	-	-	-	G1	G2	4
Online-G	G1	-	-	-	-	G1	G1	3
ONLINE-W	G2	-	-	-	-	G1	G1	4
ONLINE-Y	G2	-	-	-	-	G2	G2	6
OpenNMT	G5	-	-	-	-	-	-	5
PAHT	-	-	-	-	-	-	B1	1
PROMT	G3	-	-	-	-	G5	-	8
SPECTRANS	-	-	O4	-	-	-	-	4
SRPOL	-	-	-	-	-	G6	-	6
SRT	-	B3	-	-	-	-	-	3
super_star	-	-	-	-	-	G2	-	2
szdx	-	-	-	-	-	-	G7	7
taicangshaxigaozhong	-	-	-	-	-	-	G2	2
ustc-mt	-	-	O6	-	-	O2	O2G6	16
V2ray	-	-	-	-	-	-	G1	1
Total	40	8	26	6	6	55	114	255

Table 4: Overview of the submissions from all teams and test sets translating from English. We identify submissions using the WMT Biomedical Submission System (WMTBio) with a "B", the ones for OCELoT Biomedical Task with an "O", and the ones for OCELoT General Task with an "G". The value next to the letter indicates the number of runs for the corresponding test set, language pair, and team.

[CITE], who used the HuggingFace datasets library (Lhoest et al., 2021). Multiple tests were performed to check that the results of our evaluation scripts are comparable to those returned by OCELoT and the WMT submission system. In total, participants were allowed to upload up to 7 predictions for each sub-track.

Tables 18, 19 and 20 show the overall results of each of the three sub-tracks. Only each team's best run is presented.

6 Manual evaluation

For the MEDLINE test sets, we performed a manual evaluation for some selected runs from some of the teams. In this section we describe how the teams and runs were selected, the results of the manual evaluation, and our observations on the quality of the translations.

6.1 Selected teams and submissions

A team qualified for manual evaluation if the participants either submitted a survey or a publications with details about their submission (see Section 4). Only the following six teams complied with this requirement: ECNU-MT, Huawei-BabelTar, Huawei-TSC, SPECTRANS, SRT, and Summer.

During the submission, we asked the participants to identify a primary submission for each language pair, as indicated in Tables 10, 11, 12, 13, 14, 15, 16, and 17. For those teams who submitted runs to both submission systems, we chose the ones sent to the BioWMT submission system. The Huawei-

Teams	de2en	es2en	fr2en	it2en	pt2en	ru2en	zh2en	Total
AISP-SJTU	-	-	-	-	-	-	G1	1
ALMAnaCH-Inria	-	-	-	-	-	G2	-	2
aoligei	-	-	-	-	-	-	O4G5	9
bhcs-mt	-	-	-	-	-	-	G5	5
bymt	-	-	-	-	-	-	G1	1
ChicHealth	-	-	-	-	-	-	B3	3
Dtranx	O3G3	O1	O3	O3	O3	O2G2	O2G2	24
DLUT	-	-	-	-	-	-	G3	3
ECNU-MT	-	-	-	-	-	-	B2	2
Huawei-BabelTar	B3	B3	B3	B3	B3	B3	B3	21
Huawei-TSC	B3O4	-	B3O3	-	-	B3O4	B3O6G4	33
JDExploreAcademy.Vega-MT	G2	-	-	-	-	G3	G7	12
KwaiMT	-	-	-	-	-	-	G3	3
Lan-BridgeMT	O2G3	-	-	-	-	O2G3	O6G4	20
LanguageX	-	-	-	-	-	-	G6	6
Liaoning University	-	-	-	-	-	-	G3	3
LT22	G5	-	-	-	-	-	-	5
neunlplab	-	-	-	-	-	-	G6	6
njupt-mtt	O1	-	O3	-	-	O2G3	O2G7	18
ONLINE-A	G1	-	-	-	-	G1	G1	3
ONLINE-B	G1	-	-	-	-	G1	G1	3
Online-G	G1	-	-	-	-	G1	G1	3
ONLINE-W	G2	-	-	-	-	G1	G1	4
ONLINE-Y	G2	-	-	-	-	G2	G2	6
PAHT	-	-	-	-	-	-	B1	1
pingan_mt	-	-	-	-	-	-	G1	1
PROMT	G2	-	-	-	-	G1	-	3
SRPOL	-	-	-	-	-	G7	-	7
SPECTRANS	-	-	O4	-	-	-	-	4
SRT	-	B3	-	-	-	-	-	3
star	-	-	-	-	-	-	G4	4
super_star	-	-	-	-	-	-	G6	6
szdx	-	-	-	-	-	-	O1G7	8
Summer	-	-	-	-	-	-	B3	3
taicangshaxigaozhong	-	-	-	-	-	-	G4	4
ustc-mt	-	-	O5	-	-	O2	O1G5	13
V2ray	-	-	-	-	-	-	G1	1
Total	38	7	62	6	6	107	128	354

Table 5: Overview of the submissions from all teams and test sets translating into English. We identify submissions using the WMT Biomedical Submission System (WMTBio) with a "B", the ones for OCELoT Biomedical Task with an "O", and the ones for OCELoT General Task with an "G". The value next to the letter indicates the number of runs for the corresponding test set, language pair, and team.

Team ID	Language pair	NMT implementation	Trained	Fine-Tuned	BT	LM
ECNU_MT	en2zh, zh2en	fairseq	No	Yes	Yes	Yes
Huawei_BabelTar	en/de,es,fr,zh	fairseq	No	Yes	Yes, into en	No
Huawei_BabelTar	en/it	fairseq	No	Yes	Yes, for it2en	Yes, for it2en
Huawei_BabelTar	en/pt,ru	fairseq	No	Yes	Yes, from en	Yes, into en
Huawei_TSC	en/ru	Fairseq	No	Yes	Yes	No
Huawei_TSC	en/de, en/zh	Marian, Fairseq	No	Yes	Yes	No
Huawei_TSC	en/fr	Marian, Fairseq	Yes	No	Yes	No
SPECTRANS	en2fr	SYSTRAN Pure Neural Server 9.8	No	Yes	No	Yes
SRT	es2en	Fairseq	Yes	No	Yes	No
Summer	zh2en	Fairseq	Yes	No	Yes	No

Table 6: Overview of methods used by participating teams. Information is self-reported through the dedicated survey for each selected "best run". BT indicates if backtranslation is used and LM if language models were used.

Language pair	team	Parallel corpus	size (sentence pairs)	Monolingual corpus	size (sentences)
de/en	Huawei_BabelTar	MEDLINE corpus supplied by WMT biomedical task organizers	2.4 M	Yes	53 M (en)
	Huawei_TSC	UFAL corpus and "internal corpus"	2.75M	No	-
es/en	Huawei_BabelTar	MEDLINE corpus supplied by WMT biomedical task organizers	1.1 M	Yes	52.5 M (en)
	Huawei_TSC	corpus provided by WMT biomedical task organizers	8.1 M	Yes	8 M (en)
	SRT	MEDLINE, UFAL, MeSpEN and Scielo	3.47 M	Yes	3.5M (es), 13.9M (en)
fr/en	Huawei_BabelTar	MEDLINE corpus supplied by WMT biomedical task organizers	2.8 M	Yes	53 M (en)
	Huawei_TSC	corpus provided by WMT biomedical task organizers	6 M	Yes	2 M (en) 45M (en)
	SPECTRANS	in-house translation memory on diabetes and UFAL	2,700 (TM)	No	-
it/en	Huawei_BabelTar	MEDLINE corpus supplied by WMT biomedical task organizers	139 K	Yes	55 M (en)
pt/en	Huawei_BabelTar	MEDLINE corpus supplied by WMT biomedical task organizers	7.1 M	Yes	52.5 M (en)
en/ru	Huawei_BabelTar	Corpus supplied by WMT biomedical task organizers.	32 K	Yes	52.5 M (en)
	Huawei_TSC	Corpus supplied by organizers	24 K	Yes	46 M (en)
en/zh	ECNU_MT	NEJM en-zh corpus	66 K	Yes	40 M (en)
	Huawei_BabelTar	TAUS corpus	847 K	Yes	53 M (en)
	Huawei_TSC	UFAL and in-house corpus (unspecified)	10.87M	No	-
	Summer	MEDLINE, TAUS and covid-19 terminology by Google and Facebook	0.5 M	Yes	6.9 M (en)

Table 7: Overview of in-domain corpora used by participating teams. Information is self reported through our survey for each selected "best run" (information on the NVIDIA model is inferred from their task paper).

Language pair	team	Parallel corpus	size (sentence pairs)	Monolingual corpus	size (sentences)
en/de	Huawei_BabelTar	"in house data"	6 M	No	-
	Huawei_TSC	WMT general corpus and "internal corpus"	200 M	Yes	10M (de) 46M (en)
en/es	Huawei_BabelTar	WikiMatrix	3.3 M	No	-
	Huawei_TSC	WMT general corpus and "internal corpus"	200 M	No	-
	SRT	ParaCrawl, CommonCrawl, Europarl, News Commentary, Tatoeba, and UN Corpus	518 M	No	-
en/fr	Huawei_BabelTar	"in house corpus"	3 M	No	-
	Huawei_TSC	"in house data"	600 M	No	-
	SPECTRANS	UFAL Corpus	2,7 M	No	-
en/it	Huawei_BabelTar	"in house data"	6 M	No	-
en/pt	Huawei_BabelTar	WikiMatrix	3 M	No	-
en/ru	Huawei_BabelTar	"in house data"	3 M	No	-
	Huawei_TSC	Corpus supplied by the WMT 2022 general task	200 M	Yes	46 M (en) 40 M (ru)
en/zh	ECNU_MT	NA	-	No	-
	Huawei_BabelTar	"in house corpus"	3 M	No	-
	Huawei_TSC	"in house data"	200 M	Yes	46M (en) 92M (zh)
	Summer	Corpus supplied by the WMT 2021 News task	30.6 M	Yes	132 M (en)

Table 8: Overview of out-of-domain (OOD) corpora used by participating teams. Information is self reported through our survey for each selected "best run".

Team ID	Affiliation	Clinical Cases	Terminology	Ontology
		(en2es)	(es2en)	(en2es)
Avellana Translation	Avellana Translation	✓	✓	✓
DtranX	DtranX	✓	✓	✓
Huawei	Huawei Technologies		✓	
Logrum_UoM	University of Manchester	✓	✓	✓
Optum	Optum	✓	✓	✓

Table 9: List of the participating teams who submitted results to the ClinSpEn subtask.

Teams	Runs	en2de	en2es	en2fr	en2it	en2pt	en2ru	en2zh
ChicHealth	run1	-	-	-	-	-	-	55.71
ECNU-MT	run1	-	-	-	-	-	-	39.85
HuaweiTSC	run1	39.00	-	40.17*	-	-	41.27	50.79
	run2	39.14*	-	38.81	-	-	40.53	50.78*
	run3	38.91	-	39.00	-	-	40.63*	50.68
Huawei-BabelTar	run1	33.42	44.70	37.85	46.49	52.55	36.97	47.68
	run2	33.13	44.15	37.49	47.83	51.74	36.74	47.30
	run3	33.04	44.75	36.21	48.48	51.47	37.03	45.13
PAHT	run1	-	-	-	-	-	-	48.26
SRT	run1	-	52.14	-	-	-	-	-
	run2	-	51.96*	-	-	-	-	-
	run3	-	52.35	-	-	-	-	-
Baseline	-	29.43	39.15	28.12	47.13	42.39	27.59	39.79

Table 10: BLEU scores for "OK" aligned MEDLINE test sentences, from English, for submissions to the BioWMT Biomedical system. Primary runs are marked by *.

Teams	Runs	de2en	es2en	fr2en	it2en	pt2en	ru2en	zh2en
ChicHealth	run1	-	-	-	-	-	-	34.27
	run2	-	-	-	-	-	-	36.48
	run3	-	-	-	-	-	-	46.14*
ECNU-MT	run1	-	-	-	-	-	-	24.75*
	run2	-	-	-	-	-	-	24.49
HuaweiTSC	run1	46.95	-	50.95*	-	-	48.86	42.69
	run2	47.12*	-	50.36	-	-	50.01*	42.56*
	run3	46.82	-	50.48	-	-	49.58	42.76
Huawei-BabelTar	run1	43.10	56.60	49.08	48.83	56.03	46.16	46.12
	run2	43.75	59.02	48.86	49.16	55.44	46.26	42.49
	run3	43.38	58.64	49.36	49.89	55.63	46.75	41.80
PAHT	run1	-	-	-	-	-	-	31.16
SRT	run1	-	59.54	-	-	-	-	-
	run2	-	59.43	-	-	-	-	-
	run3	-	60.45*	-	-	-	-	-
Summer	run1	-	-	-	-	-	-	44.39*
	run2	-	-	-	-	-	-	44.31
	run3	-	-	-	-	-	-	46.17
Baseline	-	33.28	40.42	37.29	42.98	47.57	31.23	20.41

Table 11: BLEU scores for “OK” aligned MEDLINE test sentences, into English, for submissions to the BioWMT Biomedical system. Primary runs are marked by *.

BabelTar did not not indicate their primary run for some languages, and so we chose the ones with the highest scores, namely: run1 for en2de, en2fr, pt2en, and en2pt; run2 for de2en and es2en; and run3 for en2es, fr2en, it2en, en2it, ru2en, en2ru, zh2en, en2zh.

For submissions into English, we randomly selected the abstracts until we achieved at least 100 perfectly aligned (OK) sentences (see Table 2). We performed pairwise comparison between the reference translation and the selected submissions. The results from the manual validation are presented in Table 21. Unfortunately, we could not perform manual validation for submissions for de2en and

Teams	Runs	en2de	en2es	en2fr	en2it	en2pt	en2ru	en2zh
aoligei	run1	-	-	-	-	-	-	38.71
	run1	-	-	-	-	-	-	38.25*
Dtranx	run1	34.84*	49.18*	34.73	48.92	47.83	30.78*	41.14
	run2	-	49.18	35.18	47.52	37.84	17.45	36.25*
	run3	-	-	23.84*	29.20*	24.44*	-	-
Huawei-TSC	run1	34.15	-	35.02	-	-	26.88	40.25
	run2	34.04	-	34.98	-	-	30.59	40.13
	run3	34.28	-	35.56	-	-	26.73*	40.21
	run4	33.97	-	36.13*	-	-	-	39.99
	run5	34.28	-	-	-	-	-	40.12
	run6	34.28*	-	-	-	-	-	39.42
Lan-BridgeMT	run1	31.10	-	-	-	-	25.52*	37.86
	run2	31.67*	-	-	-	-	25.28	36.90
	run3	-	-	-	-	-	-	37.99
	run4	-	-	-	-	-	-	37.98*
njupt-mtt	run1	33.94	-	35.41	-	-	25.64	36.53
	run2	-	-	35.07	-	-	27.09	40.25
	run3	-	-	34.69	-	-	26.73	39.87
SPECTRANS	run1	-	-	20.68	-	-	-	-
	run2	-	-	31.63*	-	-	-	-
	run3	-	-	7.32	-	-	-	-
	run4	-	-	20.34	-	-	-	-
ustc-mt	run1	-	-	33.69	-	-	26.97	40.02
	run2	-	-	34.40	-	-	30.95	39.63
	run3	-	-	35.30	-	-	-	-
	run4	-	-	34.91	-	-	-	-
	run5	-	-	35.41	-	-	-	-
	run6	-	-	35.55	-	-	-	-

Table 12: BLEU scores for the OCELoT Biomedical Task, from English. An asterisk * indicates the primary run.

it2en.

For submissions from English, we manually selected 19 sentences from one of the clinical case reports, namely, PMID 35144678. Subsequently, we completed the sets with abstracts from the respective test sets. For the abstracts and exclusively for en2fr, for which a reference translation for the clinical case reports is available, we carry out a pairwise comparison between the reference translation and the selected submissions. For the case report for the remaining languages, we could only perform pairwise comparisons between teams' submissions. The results from the manual validation are presented in Table 22.

In both tables, we show in bold the comparisons in which one of the teams (or the reference translation) was statistically significant, according to the Wilcoxon test. The reference translation had a similar quality to many of the submissions. However, none of the teams was (statistically significant) superior than the reference translation.

6.2 Quality of the translations

Here we discuss the quality of the translations after manual validation of the selected abstracts and clinical case report.

en2fr As in previous years, the overall translation quality was high, with many automatically produced sentences exhibiting only small differences with the reference translation. In the examples shown below, correct translations are shown in black font while incorrect ones appear in red fond. Passages underlined within the same example block mark text that should carry the same meaning across statements.

(1) **en:** risk of short-term stroke
fr$_1$: risque d'AVC à court terme
fr$_2$: risque d'accident vasculaire cérébral de courte durée

(2) **en:** the long-term stroke ARD
fr$_1$: la DRA de l'AVC à long terme
fr$_2$: la maladie d'Alzheimer et les démences apparentées à long terme

However, longer or more complex sentences seemed more difficult to address for automatic systems. For examples, acronym modifiers were sometimes translated erroneously 1. We also noticed recurring issues pertaining to acronym translation (Example 2) as well as consistency throughout an entire document.

Teams	Runs	de2en	es2en	fr2en	it2en	pt2en	ru2en	zh2en
aoligei	run1	-	-	-	-	-	-	40.85
	run2	-	-	-	-	-	-	39.75
	run3	-	-	-	-	-	-	41.26*
	run4	-	-	-	-	-	-	40.24
DTranx	run1	35.55	54.21*	44.94	45.85	54.89	38.40	41.27
	run2	22.60*	-	46.38	42.04	53.67	19.34*	39.22*
	run3	37.28	-	26.91*	25.28*	28.06*	-	-
Huawei-TSC	run1	37.59	-	45.66	-	-	35.57	41.25
	run2	37.49	-	51.86	-	-	36.33	41.50*
	run3	37.62	-	46.76	-	-	35.85	41.33
	run4	37.60*	-	-	-	-	36.33*	41.33
	run5	-	-	-	-	-	-	41.66
	run6	-	-	-	-	-	-	41.46
Lan-BridgeMT	run1	35.09	-	-	-	-	31.71*	40.64
	run2	34.99*	-	-	-	-	31.24	40.09
	run3	-	-	-	-	-	-	39.78
	run4	-	-	-	-	-	-	39.31
	run5	-	-	-	-	-	-	40.73*
njupt-mtt	run1	37.09	-	45.68	-	-	35.05	41.39
	run2	-	-	44.85	-	-	35.87	41.32
	run3	-	-	44.94	-	-	-	-
SPECTRANS	run1	-	-	25.81	-	-	-	-
	run2	-	-	40.10*	-	-	-	-
	run3	-	-	25.87	-	-	-	-
	run4	-	-	9.69	-	-	-	-
ustc-mt	run1	-	-	45.11	-	-	35.39	41.05
	run2	-	-	44.81	-	-	38.48	-
	run3	-	-	45.77	-	-	-	-
	run4	-	-	45.27	-	-	-	-
	run5	-	-	0.02	-	-	-	-
szdx	-	-	-	-	-	-	-	36.00

Table 13: BLEU scores for the OCELoT Biomedical Task, into English. An asterisk * indicates the primary run.

For example, the acronym *POAF*, corresponding to the term *Perioperative atrial fibrillation*, was translated as *POAF*, *FOPA*, *FPO* or *FAPO*. Systems commonly used a combination of two or more of these solutions throughout a whole document, while the reference translation consistently used the correct translation, *FAPO*.

This year, manual validation for en2fr was performed by one evaluator with translation training and one clinician. The overall agreement on individual pair comparison was moderate at 64%. However, the overall ordering of systems and reference according to both annotator remained unchanged.

fr2en As in previous years, translation quality was high, resulting in many automatically produced translations whose quality was indistinguishable from that of reference translations. Concerning the quality of this year's references, they generally corresponded better to direct (as opposed to approximate) translations of the source abstracts, with respect to previous years. This is reflected by the pairwise comparison, which shows that the reference translation is systematically preferred over automatic translations. The most common translation errors were in term and acronym translation (Examples 3-6), prepositional and adjectival attachment (Examples 7 and 8 and in lack of capitalisation (of terms and in particular of acronyms). Term translation was particularly important for overall translation quality, often counterbalancing other more minor errors such as the naturalness of lexical and syntactic choices and correct capitalisation.

(3) **fr:** polyradiculonévrite inflammatoire démyélinisante chronique
en$_1$**:** chronic inflammatory demyelinating polyradiculoneuropathy
en$_2$**:** *chronic inflammatory demyelinating polyradiculoneuritis

(4) **fr:** défaut de croissance staturo-pondérale
en$_1$**:** failure to thrive
en$_2$**:** *staturo-weight growth defect

(5) **fr:** les inhibiteurs des cotransporteurs sodium-glucose de type 2 (iSGLT2, gliflozines)
en$_1$**:** sodium-glucose cotransporter type 2 inhibitors (SGLT2i, gliflozins)

Teams	**Runs**	**en2de**	**en2ru**	**en2zh**	
AISP-SJTU	run1	-	-	37.74	
	run2	-	-	37.70	
ALMAnaCH-Inria	run1	-	20.22	-	
	run2	-	9.77	-	
aoligei	run1	-	-	38.71	
	run2	-	-	38.25	
bhcs-mt	run1	-	-	33.61	
	run2	-	-	34.34	
	run3	-	-	39.82	
	run4	-	-	39.82	
DLUT	run1	-	-	36.22	
	run2	-	-	35.58	
	run3	-	-	36.32	
	run4	-	-	30.54	
Dtranx	run1	0.03	30.78	41.14	
	run2	34.84	17.45	37.98	
	run3	34.43	-	-	-
eTranslation	run1	-	27.53	-	
	run2	-	27.28	-	
	run3	-	27.53	-	
GTCOM	run1	-	-	38.18	
	run2	-	-	37.06	
	run3	-	-	36.94	
HuaweiTSC	run1	-	-	36.36	
	run2	-	-	35.72	
	run3	-	-	35.89	
	run4	-	-	37.95	
	run5	-	-	35.66	
	run6	-	-	37.95	
	run7	-	-	39.42	
JDExploreAcademy.Vega-MT	run1	33.32	29.77	39.24	
	run2	33.50	29.49	41.16	
	run3	-	-	41.16	
	run4	-	-	41.16	
	run5	-	-	40.40	
	run6	-	-	40.63	
	run7	-	-	39.82	
KwaiMT	run1	-	-	37.34	
	run2	-	-	41.06	
	run3	-	-	41.06	
Lan-Bridge	run1	31.10	25.52	37.86	
	run2	31.67	25.28	37.86	
	run3	31.84	25.38	36.90	
	run4	34.43	30.91	37.97	
LanguageX	run1	-	-	42.17	
	run2	-	-	41.79	
	run3	-	-	41.35	
	run4	-	-	41.57	
Manifold	run1	-	-	38.00	
	run2	-	-	38.40	
	run3	-	-	37.99	
	run4	-	-	38.10	
	run5	-	-	38.15	
	run6	-	-	38.21	
	run7	-	-	38.31	
MeteorMan	run1	-	-	38.58	
neunlplab	run1	-	-	34.76	
	run2	-	-	35.21	
	run3	-	-	35.21	
	run4	-	-	35.03	
	run5	-	-	35.14	
	run6	-	-	35.30	

Table 14: BLEU scores for OCELoT General Task, from English (part 1/2).

Teams	Runs	en2de	en2ru	en2zh
njupt-mtt	run1	-	26.43	40.25
	run2	-	27.20	36.53
	run3	-	30.95	41.16
	run4	-	25.36	37.19
	run5	-	-	37.09
	run6	-	-	37.03
	run7	-	-	37.99
ONLINE-A	run1	33.21	28.04	37.94
	run2	-	28.04	-
ONLINE-B	run1	34.88	30.90	41.17
	run2	-	-	41.17
Online-G	run1	33.76	29.68	37.31
ONLINE-W	run1	34.88	31.59	39.42
	run2	37.37	-	-
ONLINE-Y	run1	34.88	30.90	41.17
	run2	33.38	28.23	37.79
OpenNMT	run1	30.72	-	-
	run2	30.92	-	-
	run3	30.47	-	-
	run4	29.48	-	-
	run5	30.89	-	-
PROMT	run1	32.82	29.18	-
	run2	32.70	31.13	-
	run3	32.70	31.07	-
	run4	-	29.68	-
	run5	-	29.18	-
SRPOL	run1	-	27.78	-
	run2	-	27.61	-
	run3	-	27.24	-
	run4	-	27.62	-
	run5	-	27.52	-
	run6	-	27.58	-
super_star	run1	-	-	36.94
	run2	-	-	41.06
szdx	run1	-	-	38.58
	run2	-	-	38.23
	run3	-	-	38.25
	run4	-	-	38.25
	run5	-	-	38.25
	run6	-	-	38.25
	run7	-	-	38.25
taicangshaxigaozhong	run1	-	-	13.75
	run2	-	-	38.58
ustc-mt	run1	-	-	36.45
	run2	-	-	32.60
	run3	-	-	31.31
	run4	-	-	38.01
	run5	-	-	35.46
	run6	-	-	38.45
V2ray	run1	-	-	41.16

Table 15: BLEU scores for OCELoT General Task, from English (part 2/2).

en$_2$: *type 2 sodium glucose co-transporter inhibitors (iSGLT2, gliflozins)

(6) **fr:** une VCE pour OGIB en pratique courante
en$_1$: VCE for OGIB in routine practice
en$_2$: *an ECV for OGIB in current practice[10]

(7) **fr:** pour les migraines et céphalées en grappe
en$_1$: for migraines and cluster headaches
en$_2$: *for cluster migraines and headaches

(8) **fr:** les personnes non diabétiques
en$_1$: non-diabetic people
en$_2$: *non-people with diabetes

As an additional comment, some of the MT out-

[10] This example is interesting, since the original French uses English acronyms rather than French ones, presumably as they are well-known terms that have been borrowed into scientific French. The correct English translation is therefore to use the same acronyms as the French.

Teams	Runs	de2en	ru2en	zh2en
AISP-SJTU	run1	-	-	39.22
ALMAnaCH-Inria	run1	-	25.64	-
	run2	-	21.69	-
aoligei	run1	-	-	40.85
	run2	-	-	41.45
	run3	-	-	40.03
	run4	-	-	40.34
	run5	-	-	40.85
bhcs-mt	run1	-	-	31.75
	run2	-	-	39.09
	run3	-	-	39.46
	run4	-	-	40.95
	run5	-	-	41.03
bymt	run1	-	-	39.22
Dtranx	run1	35.55	38.40	41.27
	run2	37.28	19.34	39.22
	run3	22.60	-	-
DLUT	run1	-	-	33.10
	run2	-	-	32.95
	run3	-	-	33.22
HuaweiTSC	run1	-	-	36.85
	run2	-	-	34.63
	run3	-	-	36.73
	run4	-	-	36.73
JDExploreAcademy.Vega-MT	run1	35.92	37.90	39.03
	run2	36.24	37.85	40.63
	run3	-	37.90	40.73
	run4	-	-	40.48
	run5	-	-	41.14
	run6	-	-	41.41
	run7	-	-	41.27
KwaiMT	run1	-	-	41.09
	run2	-	-	39.89
	run3	-	-	39.88
Lan-Bridge	run1	35.09	31.71	40.64
	run2	34.99	31.24	40.37
	run3	35.62	38.86	40.31
	run4	-	-	40.73
LanguageX	run1	-	-	41.95
	run2	-	-	39.50
	run3	-	-	41.21
	run4	-	-	40.57
	run5	-	-	41.38
	run6	-	-	41.08
Liaoning University	run1	-	-	39.44
	run2	-	-	34.62
	run3	-	-	34.67
LT22	run1	24.69	-	-
	run2	24.59	-	-
	run3	24.22	-	-
	run4	23.19	-	-
	run5	23.19	-	-
neunlplab	run1	-	-	34.76
	run2	-	-	35.21
	run3	-	-	35.21
	run4	-	-	35.03
	run5	-	-	35.14
	run6	-	-	35.30

Table 16: BLEU scores for OCELoT General Task, into English (part 1/2).

puts appeared robust to unexpected variation in the source texts, such as rare cases of odd capitalisation, additional spaces within words and the use of inclusive writing, as can be seen in Example 5 with the word *patient.e.s* ‘patient (m/f)’, indicating the masculine and feminine forms simultaneously.

Teams	Runs	de2en	ru2en	zh2en
njupt-mtt	run1	-	35.53	34.51
	run2	-	35.66	41.38
	run3	-	33.09	34.56
	run4	-	-	35.63
	run5	-	-	36.40
	run6	-	-	0.5
	run7	-	-	35.05
ONLINE-A	run1	35.76	36.72	36.67
ONLINE-B	run1	35.50	38.27	41.03
Online-G	run1	35.30	37.69	35.88
ONLINE-W	run1	35.50	32.51	37.41
	run2	37.62	-	-
ONLINE-Y	run1	35.50	38.27	41.03
	run2	35.64	36.05	36.89
pingan_mt	run1	-	-	41.86
PROMT	run1	35.06	33.10	-
	run2	35.06	-	-
SRPOL	run1	-	33.68	-
	run2	-	34.22	-
	run3	-	33.77	-
	run4	-	34.54	-
	run5	-	34.58	-
	run6	-	34.83	-
	run7	-	34.85	-
star	run1	-	-	41.26
	run2	-	-	41.71
	run3	-	-	40.20
	run4	-	-	40.85
super_star	run1	-	-	40.42
	run2	-	-	39.48
	run3	-	-	41.07
	run4	-	-	41.59
	run5	-	-	38.80
	run6	-	-	40.85
szdx	run1	-	-	36.00
	run2	-	-	39.22
	run3	-	-	39.20
	run4	-	-	39.22
	run5	-	-	39.22
	run6	-	-	11.95
	run7	-	-	39.22
taicangshaxigaozhong	run1	-	-	39.22
	run2	-	-	39.22
	run3	-	-	14.77
	run4	-	-	39.22
ustc-mt	run1	-	-	23.17
	run2	-	-	25.00
	run3	-	-	34.96
	run4	-	-	35.71
	run5	-	-	36.68
V2ray	run1	-	-	41.17

Table 17: BLEU scores for the OCELoT General Task, into English (part 2/2).

Teams	Run	COMET	METEOR	SacreBLEU	BLEU	ROUGE
Avellana Translation	run1	0.392	0.643	36.64	35.19	0.633
DtranX	run1	0.461	0.663	**41.06**	39.36	0.649
Logrus_UoM	run1	0.423	0.633	38.17	36.50	0.627
Optum	run4	0.442	0.644	38.12	36.42	0.628

Table 18: Results for the first ClinSpEn sub-track (en2es clinical case report translation).

Teams	Run	COMET	METEOR	SacreBLEU	BLEU	ROUGE
Avellana Translation	run1	0.196	0.570	15.88	15.65	0.686
DtranX	run1	1.115	0.611	35.84	35.21	0.701
Huawei	run7	1.190	0.624	**41.57**	41.32	0.721
Logrus_UoM	run1	0.979	0.588	26.87	26.67	0.671
Optum	run2	0.982	0.574	27.94	27.57	0.656

Table 19: Results for the second ClinSpEn sub-track (es2en clinical terminology translation).

Teams	Run	COMET	METEOR	SacreBLEU	BLEU	ROUGE
Avellana Translation	run1	0.384	0.570	31.72	30.42	0.762
DtranX	run1	1.249	0.627	**58.24**	57.24	0.783
Logrus_UoM	run1	0.949	0.626	39.10	36.74	0.768
Optum	run1	1.119	0.588	44.97	43.96	0.747

Table 20: Results for the third ClinSpEn sub-track (en2es ontology concept translation).

Lang. dir.	Pair	Abstracts				Sentences			
		Total	A>B	A=B	A<B	Total	A>B	A=B	A<B
es2en	**reference** vs. Huawei-BabelTar	14	**6**	4	4	106	23	**47**	36
	reference vs. SRT	14	3	3	8	106	14	45	**47**
	Huawei-BabelTar vs. SRT	14	3	7	4	106	11	73	22
fr2en	SPECTRANS vs. Huawei-TSC	18	5	0	13	103	23	34	**46**
	SPECTRANS vs. **Huawei-BabelTar**	18	3	2	**13**	103	24	33	**46**
	SPECTRANS vs. **reference**	18	3	0	**15**	103	23	12	**68**
	Huawei-TSC vs. Huawei-BabelTar	18	11	3	4	103	40	44	**19**
	Huawei-TSC vs. reference	18	6	1	11	103	35	24	44
	Huawei-BabelTar vs. **reference**	18	2	5	**11**	103	29	22	**52**
pt2en	Huawei-BabelTar vs. reference	12	1	10	1	101	18	70	13
ru2en	reference vs. Huawei-BabelTar	14	**7**	6	1	108	31	59	18
	reference vs. Huawei-TSC	14	8	3	3	108	**44**	54	10
	Huawei-BabelTar vs. Huawei-TSC	14	1	11	2	108	7	87	14
zh2en	Summer vs. Huawei-BabelTar	17	**12**	2	3	-	-	-	-
	Summer vs. reference	17	6	7	4	-	-	-	-
	Summer vs. Huawei-TSC	17	2	11	4	-	-	-	-
	Summer vs. ECNU-MT	17	14	2	1	-	-	-	-
	Huawei-BabelTar vs. reference	17	1	9	**7**	-	-	-	-
	Huawei-BabelTar vs. Huawei-TSC	17	1	4	**12**	-	-	-	-
	Huawei-BabelTar vs. ECNU-MT	17	11	0	6	-	-	-	-
	reference vs. Huawei-TSC	17	4	9	4	-	-	-	-
	reference vs. ECNU-MT	17	**16**	1	0	-	-	-	-
	Huawei-TSC vs. ECNU-MT	17	**14**	3	0	-	-	-	-

Table 21: Pairwise manual evaluation results for the MEDLINE abstracts test set (into English). We show in bold the values which were statistically significant (Wilcoxon test). We only show the team (or reference) in bold, if both the abstracts and sentences were statistically significant (bold).

Nevertheless, most systems struggled to deal with the ambiguity linked to the translation of personal pronouns *sa*, *son*, *ses* 'his/her' in a context where it refers to an unspecified individual (e.g. *the teenager, the child*, etc.); most systems chose the masculine 'his', whereas the correct translation would either be gender neutral 'they' or 'his or her'.

From the manual evaluation results (cf. Table 21), it appears that Huawei-TSC is the superior system; although results are not significant for comparisons against the other two systems), it is the only system of the three that is not significantly worse than the reference translation. Results for abstracts and for sentences appear to correlate, although it was possible on occasions for an abstract to be of better quality than another despite having fewer better individual sentences (due to the differing importance of different errors).

Lang. dir.	**Pair**		**Abstracts**				**Sentences**		
		Total	**A>B**	**A=B**	**A<B**	**Total**	**A>B**	**A=B**	**A<B**
en2de	reference vs. Huawei-TSC	11	2	5	4	79	12	50	17
	reference vs. Huawei-BabelTar	11	**10**	0	1	79	**57**	20	2
	Huawei-TSC vs. Huawei-BabelTar	12	**9**	3	0	96	**70**	24	2
en2es	Huawei-BabelTar vs. **SRT**	11	1	2	**8**	115	11	35	**57**
	reference vs. Huawei-BabelTar	11	**7**	2	1	86	**51**	25	10
	reference vs. SRT	10	0	7	3	86	16	50	20
en2fr	**reference** vs SPECTRANS	6	**6**	0	0	87	**79**	7	0
	reference vs. Huawei-TSC	6	**6**	0	0	87	**76**	10	0
	reference vs. Huawei-BabelTar	6	**6**	0	0	87	**75**	10	1
	SPECTRANS vs. Huawei-TSC	6	1	1	4	87	27	20	40
	SPECTRANS vs. Huawei-BabelTar	6	**5**	1	0	87	**63**	18	6
	Huawei-TSC vs. Huawei-BabelTar	6	**6**	0	0	87	**59**	25	3
en2it	Huawei-BabelTar vs. reference	11	3	3	5	100	18	56	26
en2pt	reference vs. Huawei-BabelTar	6	0	6	5	105	19	54	32
en2ru	Huawei-TSC vs. Huawei-BabelTar	9	3	4	2	102	15	66	16
	Huawei-TSC vs. reference	8	3	2	3	84	14	56	13
	Huawei-BabelTar vs. reference	8	2	2	4	84	15	55	13
en2zh	Huawei-BabelTar vs. ECNU-MT	14	8	3	3	-	-	-	-
	Huawei-BabelTar vs. Huawei-TSC	14	10	1	3	-	-	-	-
	Huawei-BabelTar vs. reference	13	3	2	8	-	-	-	-
	ECNU-MT vs. Huawei-TSC	14	7	4	3	-	-	-	-
	ECNU-MT vs. reference	13	2	5	6	-	-	-	-
	Huawei-TSC vs. reference	13	2	1	**10**	-	-	-	-

Table 22: Pairwise manual evaluation results for the MEDLINE abstracts test set (from English). We show in bold the values which were statistically significant (Wilcoxon test). We only show the team (or reference) in bold, if both the abstracts and sentences were statistically significant (bold).

en2pt As shown in Table 22, the translations from the Huawei-BabelTar team achieved a similar quality as the reference translation. Similar to previous years, the translations had a good quality and we found just some few mistakes. For instance, errors in acronyms are still present, e.g. "Reforma Psiquiátrica Brasileira (RBP)" instead of "Reforma Psiquiátrica Brasileira (RPB)". Some translations might not include mistakes, but we thought that one of them was clearer than the other, e.g. "desfechos desfavoráveis tanto para a mãe quanto para o feto" (unfavorable outcomes for both mother and fetus) instead of "maus desfechos maternos e fetais" (poor maternal and fetal outcomes). Finally, we found it interesting that all query terms remained in English, namely "status epilepticus", "refractory", "treatment" and "topiramate", for both translations, in one particular sentence that discussed queries to a search tool.

pt2en As shown in Table 21, the translations from the Huawei-BabelTar team achieved a similar quality as the reference translation. The quality of both translations were usually good, but we found some differences in some situations in which we preferred one translation over the other. For instance, in cases such as "out of 100" instead of "of 100". Further, in one particular sentence, "rule out" was used as a translation for "discutir", while the other used "discuss". In many situations, we preferred translations that placed the verbs at the beginning of the sentence, such as in "We examined the absenteeism parameters..." instead of at the end, such as in "the parameters for granting time off work .... were analyzed". Further, we find that the use of a specific and more suitable terms, such as "absenteeism", "productivity", and "control" are preferred to a longer or informal expression, such as "granting time off work", "being productive" and "combat", respectively.

en2es This year the overall quality of the translations was mixed. Both SRT and reference translations were of very good quality, and SRT was in many occasions indistinguishable to reference translations in the manual evaluation when it came to quality. However, the Huawei-BabelTar system had a mixed result with very good translations and translations of doubtful quality that clearly affected the fluency and readability of the output.

Capitalization and word separation were the main issues encountered when evaluating Huawei-BabelTar's output at a sentence level e.g. "La pandemia de covid-19 ofreció a la humanidad un portal a través del cual podemos romper con el pasado e imaginar nuestromundo de nuevo."

As in past years, the translation of acronyms and out-of-dictionary terminology remains a challenge for MT systems, Huwaei-BabelTar being a perfect example of such issues: "Describimos el caso de una cirrosis descompensada que desarrolló hpp y se resolvió con trasplante hepático, permaneciendo asintomática tras diez años de seguimiento."

When dealing with long named entities, word order remained a challenge for both SRT and Huawei-BabelTar, as in the following example where the numbers relate to the acronym "MMPs", and not to the noun "haplotypes":

(9) **Source**: To evaluate MMPs 7, 8, 12, and 13 haplotypes and their association with CRC.

Reference: Evaluar haplotipos de las MMP 7, 8, 12, y 13 y su asociación con CCR.

Huawei-BabelTar: Evaluar los haplotipos 7,8, 12 y 13 (incorrect word order and word separation) delmmp (word separation and capitalization) y su asociación con el ccr (capitalization of acronyms).

SRT: Evaluar los haplotipos 7, 8, 12 y 13 (incorrect word order) de MMP y su asociación con el CCR.

The reference translations were of very high quality overall, creating readable and fluent outputs at the level of sentences and abstracts. The main thing that differentiated reference translations from machine translations was the fact that they were less literal and followed writing conventions in Spanish for the domain, such as the use of passive reflexive tense which is more common in Spanish medical and scientific writings. However, the reference translation omitted relevant information or added implicit information from the text, which affect the overall quality of those translations when compared with the MT systems.

es2en This year the overall quality of the translations was mixed, with some very good quality translations coming from the MT systems (which made them nearly indistinguishable from the reference translations) to poorly written translations (including reference translations). Such is the case as well for the source text, which included some very high quality abstracts and also some poorly written abstractsn which contained grammatical errors such as lack of capitalization, wrong punctuation or word separation as in the following example:

(10) estudio observacional, relacional, transversal, en 185 derechohabientes de una unidad de medicina familiar del 15 de junio al 15 de agosto de 2020

This affected the quality of the output of both MT systems, Huwaei-BabelTar and SRT, which closely followed the source text:

(11) **Huawei-BabelTar**: observational, relational, cross-sectional study in 185 beneficiaries of a family medicine unit from June 15 to August 15, 2020.

SRT: observational, relational, cross-sectional study in 185 beneficiaries of a family medicine unit from June 15 to August 15, 2020.

On the other hand, SRT proved to be more robst than Huwaei-BabelTar and the reference translation, and was able to deal with poor source text much more consistently such as in the example:

(12) **Source**: Existen múltiples causas delesiones ureterales, siendo la principal yatrogénica.

SRT: There are multiple causes of uretral injuries, the main one being iatrogenic.

Word order in longer sentences still remains a challenge for MT systems, which do not always correctly identify adverbs modifying long named entities as seen the following example, where "muchos" modifies the noun "biomarcadores":

(13) **Source** : La expansión y el descubrimiento de nuevas posibilidades de diagnóstico para el uso de muchos biomarcadores de enfermedades cardiovasculares (ECV), incluidas las isoformas de troponina cardioespecíficas (cTnI, cTnT), se debe a la mejora de los métodos de laboratorio para su determinación.

Huawei-BabelTar: The expansion and discovery of new diagnostic possibilities for the use of many cardiovascular disease (CVD) biomarkers, including cardio-specific troponin isoforms (cTnI, cTnT), is due to improved laboratory methods for their determination.

> **SRT**: The expansion and discovery of new diagnostic possibilities for the use of many cardiovascular disease (CVD) biomarkers, including cardio-specific troponin isoforms (cTnI, cTnT), is due to improved laboratory methods for their determination.

Both SRT and Huawei-BabelTar create sentences where “many” modifies “cardiovascular diseases”, which changes the meaning of the translation in both cases.

However, the reference translations also had a mixed quality when compared to the MT systems, and presented issues such as poor capitalization or incorrect word separation, as seen in the following example: “There are many causesof ureteral injury being the main one iatrogenic”.

Unlike previous years, SRT performed best in the three-way manual evaluation, coming close to the reference translation, due to the references’ varied quality.

en2de Similarly to the last few years, the quality of the translations into German was very high. Both participants provided mostly convincing translations - partially including slight restructurings of the sentences. However, although the Huawei-BabelTar team performed lower in comparison to Huawei-TSC, the translations were in most cases not necessarily of lower quality. Instead, the Huawei-BabelTar system made two crucial errors, namely a) translations tend to ignore the capitalization of some German words, as well as b) single words were sometimes written together (without whitespace). Without those two error patterns, the quality of both translation systems would be closer to each other. Sometimes the systems used literal translatations, which impacted the quality of the translated text. For instance, “real-data” was translated into “reale Daten” (instead of “Daten aus der Praxis”) or “essential” was translated into “essentiell” instead of “unerlässlich”.

en2zh The translation quality this year was high. Unlike last year where few sentences were so awkwardly translated that a reader could hardly guess the original meaning, there were essentially no such sentences this year.

The biggest factor that reduced translation quality was the treatment of biomedical terms. This phenomenon came in two categories. The first category was straightforward, where the correct Chinese term was imprecise or downright wrong. For instance, *poor outcomes* of medical treatments was imprecisely translated as 不良结局 (poor end results), when the precise Chinese medical term was 不良预后 . In another example, *Rights-based Approaches (RBAs)* was translated as 基于权利的方法（非洲区域局） in which the full name of the term was correctly translated, but the abbreviation in brackets was incorrectly translated as *Regional Bureau Africa*.

The second category is more subtle, where the translated Chinese term was correct, but the presence of the original English term (or lack thereof) impacted readability. As an example, *Diabetic Retinopathy (DR)* was ideally translated as 糖尿病视网膜病变 *(Diabetic Retinopathy, DR)*, where the Chinese term, the English term, as well as the English abbreviation in the source text were all present. Another translation omitted the full English term, yielding 糖尿病视网膜病变 *(DR)*, which was still easily understandable. In another case, however, *healthcare workers (HCWs)* was translated to 医护人员 *(HCW)*. Here, the abbreviation was translated in singular form, conflicting with the plural form in the source text.

An interesting observation was that conventional, typical wording and punctuation in the translation significantly improved its quality. As a simple example, *experts disagreed* was translated by one system as 专家意见有分歧 (expert opinions differ) and by another as 专家们持不同看法 (experts have different opinions) . Both translations conveyed the same information, but the first translation was much more typical – refined even, as one would expect in a scientific publication. In terms of punctuation, the 、 is unique to the Chinese language when listing items. Hence when given *three overall reactions (positive, negative, and ambivalent)*, the translation 三种总体反应（积极、消极和矛盾） (note the punctuation between the first and second items) read much more naturally than 三种总体反应（积极，消极和矛盾） (exactly the same text, but a comma was used instead). In these cases, the less typical writing style was strictly speaking not wrong, but immediately hinted at the possibility that the text was not written by a native speaker.

Finally, the conversion between Western and Chinese number systems remained a challenge for some systems. The amount *598.851 billion yuan* referred to a billion as 10^9. The closest Chinese word to *billion* is 亿 , which is one order of magnitude smaller at 10^8. This particular amount (598,851,000,000) was incorrectly converted to *598851* 亿元 (59,885,100,000,000) by one system, and correctly though confusingly converted to *558851 m* 元 (558,851 million) by another.

zh2en Continuing the trend from the previous two years, the translations this year are again of high quality. Nevertheless, a few common types of error still provide room for improvement.

Presumably, when a technical or medical term is missing from the system's dictionary, the individual Chinese characters in the term are translated literally. For instance, 清零 (zero-COVID policy) was translated by multiple systems as *zeroing*, which, despite the context of a COVID-related abstract, was hardly guessable. In another instance, 增强现实 (augmented reality) is arguably a technical term outside of the biomedical domain, but was still successfully translated as *augmented reality* by most systems and only one system produced *augmented real-world*.

In other cases, when a Chinese word has a general, non-biomedical meaning as well as a biomedical one, a system might incorrectly opt for the biomedical meaning. 服务阵地不断萎缩 (the continuous shrinking/decline of the service locations) is an example, where 萎缩 should be given the general translation of *decline* instead of the biomedical translation of *atrophy*.

When a translation overly preserves the fidelity of the source phrase, the resulting translation can be awkward. Take 在明确针刺可调节神经、血管这一共识的基础上 as an example. A more readable and thus preferable translation was *based on the consensus that acupuncture can regulate nerves and blood vessels*, even though a word-for-word translation would produce *on the basis of the consensus that acupuncture can regulate nerves and blood vessels* instead.

Similar to en2zh translations, numerical values also proved challenging for some systems in zh2en. 4.26 万 (one 万 is 10,000) is equivalent to 42,600, but the systems translated that variously to *4.26,000* or even *426 million*.

6.3 Targeted evaluation in clinical cases

This year, special attention was given to the evaluation of translations submitted by systems for the clinical case reports, from English into French.

The manual evaluation focused on the criteria that were used to select the clinical case: (1) acronyms; (2) numeric values including lab values; and (3) clinical correctness. Examples 14 and 16 illustrate erroneous translations produced by automatic systems while example 15 illustrates a case of an untranslated value. In the examples correct translations are shown in black font while incorrect ones appear in red fond. An asterisk indicates ungrammatical segments. Passages underlined within the same example block mark text that should carry the same meaning across statements.

(14) **en:** screening test for SARS-CoV-2
fr$_1$: dépistage systématique du SARS-CoV-2
en$_2$: *dépistage systématique du CoV-2 du SARS

(15) **en:** the platelet count was 113 × 10E9/L
fr$_1$: les plaquettes sont à 113 000/mm^3
en$_2$: numération plaquettaire de 113 10E09/L

(16) **en:** General examination revealed a wasted man
fr$_1$: L'examen clinique objective une dénutrition
en$_2$: L'examen général a révélé un homme obèse

Specifically, the translations were annotated using BRAT[11] and aimed to assess the systems' performance on the specific aims. Annotations were produced independently by one annotator with formal translation training (AN) and one clinician (CG). For annotations on the full-text case descriptions, inter-annotator agreement on entities was high overall (above 0.75 F-measure) for "values" and "acronyms" and lower for "errors" (above 0.35 F-measure), mainly due to the identification of more errors by the clinician, which was expected. Inter-annotator agreement on attributes was medium overall (above 0.55 F-measure) mainly due to disagreements on "unclear" and "erroneous" translations, while agreement was much higher for "correct" and "untranslated" cases.

While the "correct" translation category was the most prevalent for all systems for values and acronyms, it can be noted that SPECTRANS produced more "Untranslated" occurrences. Overall, Huawei-BabelTar produced more "Errors" than the other two systems.

This analysis suggests that, in spite of high BLEU scores, the automatic translations can contain serious translation errors (e.g. Example 16)

[11] Brat Rapid Annotation Tool https://brat.nlplab.org/(Stenetorp et al., 2012)

and information that is not directly actionable for clinicians (e.g. Example 15).

7 Conclusions

We presented an overview of this year's edition of the WMT Biomedical Task. We released test sets for seven language pairs, and addressed a variety of textual sources, such as scientific abstracts, clinical case reports, and terminologies. We had a record number of participating teams and of submissions. All submissions were automatically evaluated in terms of BLEU scores, with respect to reference translations, whenever available. We also manually evaluated a selection of the submissions, and similar to previous years, the translations from some teams achieved a similar quality to the reference translations.

Limitations

The scope of the biomedical task has been growing over the years. While each new edition builds on the experience of the previous one, the scale of operations implies a number of limitations from operational and theoretic perspectives. One major limitation is the comparison between translation approaches used by the teams. The information we collect through the participant survey attempts to document the material and methods used by the participants' systems. However, it can be noted that only a subset of teams do supply details of their systems. Furthermore, some descriptions such as the training corpus size or content could be clarified. A closed task, where all participants are limited to using specific training material, could help improve comparability but would require additional work from participants and organizers. Another limitation is the imbalance between language pairs, which attracts different levels of effort from both participants and organizers.

MT can be computationally intensive and the environmental impact of experiments should be measured. While no measure of impact was conducted this year, we included this aspect in the participant survey, which included a list of tools that can be used to measure impact. A future growth direction to increase awareness of impact can be to ask participants to supply a measure of CO2 impact along with their results.

Ethics Statement

This task mainly focuses on translation using the MEDLINE corpus, which is openly available for research. The test corpora used in the task were selected based on publication date and linguistic criteria. Any imbalance regarding the demographics of populations represented in the corpus is involuntary.

The intended use of this task is to contribute to the evaluation and training of MT systems in the biomedical domain. We do not recommend the use of MT without expert validation in a medical context, as machine translated text could contain errors impacting patients' health outcomes.

Acknowledgements

The ClinSpen sub-task was supported by the Spanish Government's Encargo of Plan TL (SEAD) to BSC and BIOMATDB project of European Union's Horizon Europe Coordination & Support Action under Grant Agreement No 101058779, as well as AI4PROFHEALTH project (PID2020-119266RA-I00).

Rachel Bawden's participation was funded by her chair position in the PRAIRIE institute funded by the French national agency ANR as part of the "Investissements d'avenir" programme under the reference ANR-19-P3IA-0001.

References

Nicolas Ballier, Jean-Baptiste Yunès, Guillaume Wisniewski, Lichao Zhu, and Maria Zimina. 2022. The SPECTRANS System Description for the WMT22 Biomedical Task. In *Proceedings of the Seventh Conference on Machine Translation: Shared Task Papers*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Nesrine Bannour, Sahar Ghannay, Aurélie Névéol, and Anne-Laure Ligozat. 2021. Evaluating the carbon footprint of nlp methods: a survey and analysis of existing tools. In *EMNLP, Workshop SustaiNLP*.

Rachel Bawden, Kevin Bretonnel Cohen, Cristian Grozea, Antonio Jimeno Yepes, Madeleine Kittner, Martin Krallinger, Nancy Mah, Aurelie Neveol, Mariana Neves, Felipe Soares, Amy Siu, Karin Verspoor, and Maika Vicente Navarro. 2019. Findings of the WMT 2019 biomedical translation shared task: Evaluation for MEDLINE abstracts and biomedical terminologies. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 29–53, Florence, Italy. Association for Computational Linguistics.

Rachel Bawden, Giorgio Maria Di Nunzio, Cristian Grozea, Inigo Jauregi Unanue, Antonio Jimeno Yepes, Nancy Mah, David Martinez, Aurélie Névéol, Mariana Neves, Maite Oronoz, Olatz Perez-de Viñaspre, Massimo Piccardi, Roland Roller, Amy Siu, Philippe Thomas, Federica Vezzani, Maika Vicente Navarro, Dina Wiemann, and Lana Yeganova. 2020. Findings of the WMT 2020 biomedical translation shared task: Basque, Italian and Russian as new additional languages. In *Proceedings of the Fifth Conference on Machine Translation*, pages 660–687, Online. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Yoonjung Choi, Jiho Shin, Yonghyun Ryu, and Sangha Kim. 2022. SRT's Neural Machine Translation System for WMT22 Biomedical Translation Task. In *Proceedings of the Seventh Conference on Machine Translation: Shared Task Papers*.

Christian Federmann. 2010. Appraise: An Open-Source Toolkit for Manual Phrase-Based Evaluation of Translations. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*, pages 1731–1734, Valletta, Malta.

Antonio Jimeno Yepes, Aurelie Neveol, Mariana Neves, Karin Verspoor, Ondrej Bojar, Arthur Boyer, Cristian Grozea, Barry Haddow, Madeleine Kittner, Yvonne Lichtblau, Pavel Pecina, Roland Roller, Rudolf Rosa, Amy Siu, Philippe Thomas, and Saskia Trescher. 2017. Findings of the WMT 2017 Biomedical Translation Shared Task. In *Proceedings of the Second Conference on Machine Translation*, pages 234–247, Copenhagen, Denmark.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ernan Li, Fandong Meng, and Jie Zhou. 2022. Summer: WeChat Neural Machine Translation Systems for the WMT22 Biomedical Translation Task. In *Proceedings of the Seventh Conference on Machine Translation: Shared Task Papers*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Boxiang Liu and Liang Huang. 2021. Paramed: a parallel corpus for english–chinese translation in the biomedical domain. *BMC Medical Informatics and Decision Making*, 21(1):1–11.

Mariana Neves, Antonio Jimeno Yepes, Aurélie Névéol, Cristian Grozea, Amy Siu, Madeleine Kittner, and Karin Verspoor. 2018. Findings of the WMT 2018 Biomedical Translation Shared Task: Evaluation on MEDLINE test sets. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 324–339. Association for Computational Linguistics.

Aurélie Névéol, Antonio Jimeno Yepes, Mariana Neves, and Karin Verspoor. 2018. Parallel Corpora for the Biomedical Domain. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adrien Pavao, Isabelle Guyon, Anne-Catherine Letournel, Xavier Baró, Hugo Escalante, Sergio Escalera, Tyler Thomas, and Zhen Xu. 2022. Codalab competitions: An open source platform to organize scientific challenges. *Technical report*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on*

Machine Translation: Research Papers, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107.

Weixuan Wang, Xupeng Meng, Suqing Yan, Ye TIAN, and Wei Peng. 2022. Huawei BabelTar NMT at WMT22 Biomedical Translation Task: How we further improve domain-specific NMT. In *Proceedings of the Seventh Conference on Machine Translation: Shared Task Papers*.

Zhanglin Wu, Jinlong Yang, Zhiqiang Rao, Zhengzhe Yu, Daimeng Wei, Xiaoyu Chen, Zongyao Li, Hengchao Shang, Shaojun Li, Ming Zhu, Yuanchang Luo, Yuhao Xie, Miaomiao Ma, Ting Zhu, Lizhi Lei, Song Peng, Hao Yang, and Ying Qin. 2022. HW-TSC Translation Systems for the WMT22 Biomedical Translation Task. In *Proceedings of the Seventh Conference on Machine Translation: Shared Task Papers*.

Lana Yeganova, Dina Wiemann, Mariana Neves, Federica Vezzani, Amy Siu, Inigo Jauregi Unanue, Maite Oronoz, Nancy Mah, Aurélie Névéol, David Martinez, Rachel Bawden, Giorgio Maria Di Nunzio, Roland Roller, Philippe Thomas, Cristian Grozea, Olatz Perez-de Viñaspre, Maika Vicente Navarro, and Antonio Jimeno Yepes. 2021. Findings of the WMT 2021 biomedical translation shared task: Summaries of animal experiments as new test set. In *Proceedings of the Sixth Conference on Machine Translation*, pages 664–683, Online. Association for Computational Linguistics.

Huanran Zheng, Wei Zhu, and Xiaoling Wang. 2022. ECNU-MT's WMT22 Biomedical Translation Task Submission. In *Proceedings of the Seventh Conference on Machine Translation: Shared Task Papers*.

A Mapping of runs in OCELoT

Teams	Runs	en2de	en2es	en2fr	en2it	en2pt	en2ru	en2zh
aoligei	run1	-	-	-	-	-	-	24
	run1	-	-	-	-	-	-	25*
Dtranx	run1	360*	283*	277	304	305	352*	355*
	run2	-	291	416	418	421	469	379
	run3	-	-	448*	449*	450*	-	-
Huawei-TSC	run1	251	-	517	-	-	659	249
	run2	395	-	534	-	-	663	396
	run3	480	-	645	-	-	671*	478
	run4	520	-	774*	-	-	-	536
	run5	832	-	-	-	-	-	636
	run6	837*	-	-	-	-	-	778
Lan-BridgeMT	run1	115	-	-	-	-	113*	9
	run2	201*	-	-	-	-	198	177
	run3	-	-	-	-	-	-	202
	run4	-	-	-	-	-	-	388*
njupt-mtt	run1	140	-	124	-	-	142	88
	run2	-	-	128	-	-	155	90
	run3	-	-	579	-	-	163	93
SPECTRANS	run1	-	-	398	-	-	-	-
	run2	-	-	460*	-	-	-	-
	run3	-	-	484	-	-	-	-
	run4	-	-	486	-	-	-	-
ustc-mt	run1	-	-	312	-	-	345	722
	run2	-	-	314	-	-	369	764
	run3	-	-	542	-	-	-	-
	run4	-	-	543	-	-	-	-
	run5	-	-	544	-	-	-	-
	run6	-	-	569	-	-	-	-

Table 23: Mapping of the MEDLINE runs to the submission ids in OCELoT Biomedical Task, from English. An asterisk * indicates the primary run.

Teams	Runs	de2en	es2en	fr2en	it2en	pt2en	ru2en	zh2en
aoligei	run1	-	-	-	-	-	-	17
	run2	-	-	-	-	-	-	20
	run3	-	-	-	-	-	-	21*
	run4	-	-	-	-	-	-	23
DTranx	run1	357	303*	306	307	308	353	356
	run2	472*	-	422	423	425	470*	471*
	run3	473	-	451*	452*	453*	-	-
Huawei-TSC	run1	252	-	646	-	-	596	250
	run2	411	-	732	-	-	597	397*
	run3	481	-	750	-	-	600	479
	run4	537*	-	-	-	-	601*	523
	run5	-	-	-	-	-	-	638
	run6	-	-	-	-	-	-	781
Lan-BridgeMT	run1	104	-	-	-	-	105*	19
	run2	200*	-	-	-	-	199	203
	run3	-	-	-	-	-	-	220
	run4	-	-	-	-	-	-	221
	run5	-	-	-	-	-	-	387*
njupt-mtt	run1	139	-	125	-	-	145	89
	run2	-	-	129	-	-	156	95
	run3	-	-	580	-	-	-	-
SPECTRANS	run1	-	-	399	-	-	-	-
	run2	-	-	462*	-	-	-	-
	run3	-	-	487	-	-	-	-
	run4	-	-	492	-	-	-	-
ustc-mt	run1	-	-	316	-	-	346	724
	run2	-	-	317	-	-	371	-
	run3	-	-	565	-	-	-	-
	run4	-	-	567	-	-	-	-
	run5	-	-	568	-	-	-	-
szdx	-	-	-	-	-	-	-	97

Table 24: Mapping of the runs to the submission ids in OCELoT Biomedical task, into English. An asterisk * indicates the primary run.

Teams	Runs	en2de	en2ru	en2zh
AISP-SJTU	run1	-	-	31
	run2	-	-	611
ALMAnaCH-Inria	run1	-	381	-
	run2	-	711	-
aoligei	run1	-	-	26
	run2	-	-	27
bhcs-mt	run1	-	-	43
	run2	-	-	44
	run3	-	-	170
	run4	-	-	172
DLUT	run1	-	-	430
	run2	-	-	649
	run3	-	-	651
	run4	-	-	721
Dtranx	run1	319	329	333
	run2	325	461	354
	run3	765	-	-
eTranslation	run1	-	337	-
	run2	-	339	-
	run3	-	341	-
GTCOM	run1	-	-	521
	run2	-	-	733
	run3	-	-	853
HuaweiTSC	run1	-	-	236
	run2	-	-	465
	run3	-	-	476
	run4	-	-	557
	run5	-	-	575
	run6	-	-	630
	run7	-	-	776
JDExploreAcademy.Vega-MT	run1	507	509	59
	run2	843	690	98
	run3	-	-	102
	run4	-	-	833
	run5	-	-	652
	run6	-	-	706
	run7	-	-	834
KwaiMT	run1	-	-	794
	run2	-	-	797
	run3	-	-	799
Lan-Bridge	run1	114	112	12
	run2	191	197	162
	run3	393	409	175
	run4	549	556	714
LanguageX	run1	-	-	150
	run2	-	-	692
	run3	-	-	701
	run4	-	-	716
Manifold	run1	-	-	28
	run2	-	-	136
	run3	-	-	231
	run4	-	-	336
	run5	-	-	440
	run6	-	-	604
	run7	-	-	820
MeteorMan	run1	-	-	230
neunlplab	run1	-	-	14
	run2	-	-	67
	run3	-	-	570
	run4	-	-	760
	run5	-	-	798
	run6	-	-	847

Table 25: Mapping of the runs to the submission ids in OCELoT General Task, from English (part 1/2).

Teams	**Runs**	**en2de**	**en2ru**	**en2zh**
njupt-mtt	run1	-	137	85
	run2	-	147	92
	run3	-	213	144
	run4	-	243	211
	run5	-	-	214
	run6	-	-	216
	run7	-	-	232
ONLINE-A	run1	901	912	914
	run2	-	911	-
ONLINE-B	run1	920	930	931
	run2	-	-	932
Online-G	run1	865	876	878
ONLINE-W	run1	954	966	968
	run2	959	-	-
ONLINE-Y	run1	939	949	951
	run2	973	983	985
OpenNMT	run1	207	-	-
	run2	210	-	-
	run3	321	-	-
	run4	493	-	-
	run5	746	-	-
PROMT	run1	68	42	-
	run2	334	71	-
	run3	694	72	-
	run4	-	73	-
	run5	-	804	-
SRPOL	run1	-	157	-
	run2	-	160	-
	run3	-	265	-
	run4	-	496	-
	run5	-	497	-
	run6	-	501	-
super_star	run1	-	-	228
	run2	-	-	229
szdx	run1	-	-	119
	run2	-	-	338
	run3	-	-	436
	run4	-	-	438
	run5	-	-	439
	run6	-	-	441
	run7	-	-	442
taicangshaxigaozhong	run1	-	-	788
	run2	-	-	811
ustc-mt	run1	-	-	276
	run2	-	-	279
	run3	-	-	281
	run4	-	-	293
	run5	-	-	328
	run6	-	-	373
V2ray	run1	-	-	47

Table 26: Mapping of the runs to the submission ids in OCELoT General Task, from English (part 2/2).

Teams	**Runs**	**de2en**	**ru2en**	**zh2en**
AISP-SJTU	-	-	-	648
ALMAnaCH-Inria	run1	-	382	-
	run2	-	710	-
aoligei	run1	-	-	11
	run2	-	-	146
	run3	-	-	151
	run4	-	-	154
	run5	-	-	295
bhcs-mt	run1	-	-	45
	run2	-	-	171
	run3	-	-	173
	run4	-	-	737
	run5	-	-	810
bymt	run1	-	-	294
Dtranx	run1	315	343	349
	run2	429	463	468
	run3	456	-	-
DLUT	run1	-	-	432
	run2	-	-	653
	run3	-	-	654
HuaweiTSC	run1	-	-	245
	run2	-	-	467
	run3	-	-	571
	run4	-	-	626
JDExploreAcademy.Vega-MT	run1	508	510	58
	run2	809	769	99
	run3	-	844	101
	run4	-	-	656
	run5	-	-	658
	run6	-	-	708
	run7	-	-	736
KwaiMT	run1	-	-	415
	run2	-	-	790
	run3	-	-	792
Lan-Bridge	run1	103	86	10
	run2	188	187	222
	run3	587	589	223
	run4	-	-	386
LanguageX	run1	-	-	168
	run2	-	-	218
	run3	-	-	219
	run4	-	-	400
	run5	-	-	412
	run6	-	-	417
Liaoning University	run1	-	-	152
	run2	-	-	498
	run3	-	-	830
LT22	run1	605	-	-
	run2	608	-	-
	run3	612	-	-
	run4	614	-	-
	run5	617	-	-
neunlplab	run1	-	-	14
	run2	-	-	67
	run3	-	-	570
	run4	-	-	760
	run5	-	-	798
	run6	-	-	847

Table 27: Mapping of the runs to the submission ids in OCELoT General Task, into English (part 1/2).

Teams	Runs	de2en	ru2en	zh2en
njupt-mtt	run1	-	138	87
	run2	-	153	143
	run3	-	254	212
	run4	-	-	215
	run5	-	-	217
	run6	-	-	237
	run7	-	-	244
ONLINE-A	run1	903	913	915
ONLINE-B	run1	923	934	935
Online-G	run1	868	861	879
ONLINE-W	run1	956	967	969
	run2	961	-	-
ONLINE-Y	run1	941	950	952
	run2	975	984	986
pingan_mt	-	-	-	494
PROMT	run1	29	70	-
	run2	796	-	-
SRPOL	run1	-	272	-
	run2	-	359	-
	run3	-	361	-
	run4	-	661	-
	run5	-	664	-
	run6	-	666	-
	run7	-	697	-
star	run1	-	-	296
	run2	-	-	297
	run3	-	-	602
	run4	-	-	665
super_star	run1	-	-	159
	run2	-	-	166
	run3	-	-	165
	run4	-	-	167
	run5	-	-	227
	run6	-	-	242
	run2	-	-	166
	run3	-	-	165
	run4	-	-	167
	run5	-	-	227
	run6	-	-	242
szdx	run1	-	-	100
	run2	-	-	123
	run3	-	-	134
	run4	-	-	599
	run5	-	-	631
	run6	-	-	634
	run7	-	-	635
taicangshaxigaozhong	run1	-	-	618
	run2	-	-	640
	run3	-	-	791
	run4	-	-	813
ustc-mt	run1	-	-	280
	run2	-	-	282
	run3	-	-	292
	run4	-	-	327
	run5	-	-	477
V2ray	run1	-	-	48

Table 28: Mapping of the runs to the submission ids in OCELoT General Task, into English (part 2/2).

Findings of the WMT 2022 Shared Task on Chat Translation

Ana C. Farinha[1*] **M. Amin Farajian**[1*]
Marianna Buchicchio[1] **Patrick Fernandes**[4,5,6] **José G. C. de Souza**[1]
Helena Moniz[1,2,3] **André F. T. Martins**[1,4,5]

[1]Unbabel, Lisbon, Portugal [2]INESC-ID, Lisbon, Portugal
[3]Faculdade de Letras, University of Lisbon, Portugal
[4]Instituto de Telecomunicações, Lisbon, Portugal
[5]Instituto Superior Técnico, University of Lisbon, Portugal
[6]Carnegie Mellon University, Pittsburgh, USA

Abstract

This paper reports the findings of the second edition of the Chat Translation Shared Task. Similarly to the previous WMT 2020 edition, the task consisted of translating bilingual customer support conversational text. However, unlike the previous edition, in which the bilingual data was created from a synthetic monolingual English corpus, this year we used a portion of the newly released Unbabel's MAIA corpus, which contains genuine bilingual conversations between agents and customers. We also expanded the language pairs to English↔German (en↔de), English↔French (en↔fr), and English↔Brazilian Portuguese (en↔pt-br).

Given that the main goal of the shared task is to translate bilingual conversations, participants were encouraged to train and test their models specifically for this environment. In total, we received 18 submissions from 4 different teams. All teams participated in both directions of en↔de. One of the teams also participated in en↔fr and en↔pt-br. We evaluated the submissions with automatic metrics as well as human judgments via Multidimensional Quality Metrics (MQM) on both directions. The official ranking of the systems is based on the overall MQM scores of the participating systems on both directions, i.e. *agent* and *customer*.

1 Introduction

With the significant translation quality improvements brought by newer machine translation (MT) approaches in the last years, we can start using MT to translate non-conventional content types such as bilingual and multilingual conversations. These new applications pose new challenges to MT systems and require new solutions to deal with them. In this shared task, we focus on the automatic translation of conversational text, in particular customer support chats, an important and challenging content due to their particular characteristics (Gonçalves et al., 2022; Wang et al., 2021; Farajian et al., 2020): In contrast to content types such as news articles and software manuals, among others, in which the text is carefully authored and well formatted, chat conversations are less planned, more informal, and often present ungrammatical linguistic structures. Furthermore, such conversations are usually on-the-fly production of text with very fuzzy frontiers with speech and mimicking speech production. Due to time requirements, since in dialogues turn-exchange need to be dynamic, the conversations may also have typos, abbreviations and ellipses. The conversations are also characterized by stressful moments, which in turn is represented by the capitalization of the entire word or turn, emoticons or emojis, and multiple punctuation marks.

Furthermore, Gonçalves et al. describe several factors that often lead to poor quality of the written text in this content type, resulting in lower quality of the MT outputs. They highlight the fact that the clients requiring customer support usually demonstrate high levels of impatience and frustration, resulting in typos, profanities, as well as variable capitalization and punctuation. They also mention that text is often times left unstructured, informal and agrammatical, factors that further increase the challenges of dealing with this particular content.

Given the limited number of parallel data for this domain, the main motivation for the Chat Translation Shared Task is to provide a common ground for evaluating and analyzing the challenges posed by conversational data as a content type, which has broad application in industry-level services. Following the success of the first edition of the Chat Translation Shared Task (Farajian et al., 2020), this

[*]These authors contributed equally.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 724 - 743
December 7-8, 2022 ©2022 Association for Computational Linguistics

customer	source_segment: target_segment:	Ola, tudo bem? Hello! How are you?
customer	source_segment: target_segment:	Alguns meses atras, precisei restaurar o aplicativo da #PRS_ORG# para PC. A few months ago, I needed to restore the #PRS_ORG# PC App.
customer	source_segment: target_segment:	Quando fiz isso, perdi todos os meus livros comprados. When I did that, I lost all my purchased books.
customer	source_segment: target_segment:	Gostaria de saber como recupera-los. I would like to know how to recover them.
customer	source_segment: target_segment:	Obrigada. Thank you.
customer	source_segment: target_segment:	Celular para contato: #PHONENUMBER#. Mobile for contact: #PHONENUMBER#
agent	source_segment: target_segment:	Thank you for the information. Agradeço pela informação.
agent	source_segment: target_segment:	I will be more than happy to assist you. Terei todo o prazer em ajudar você.
agent	source_segment: target_segment:	I see all your books are in the account linked to the #EMAIL# Vejo que todos os seus livros estão na conta vinculada ao #EMAIL#

Table 1: Excerpt of a en↔pt-br conversation between a *customer* and an *agent*.

year we organized the second edition of the task with the following improvements:

- We released a genuine bilingual corpus, the Unbabel's MAIA Dataset. This consists of customer support dialogues in which the speakers (i.e. *customer* and *agent*) speak in their own language.

- We expanded the number of language pairs to three: English-German (en↔de), English-French (en↔fr), and English-Brazilian Portuguese (en↔pt-br).

- We performed manual evaluation on both directions of agent and customer, and we ranked the systems based on their overall performance in both directions, by using an adaptation of the multidimensional quality metrics (MQM) (Lommel et al., 2014) that is tailored to assess customer support translated content.

Similarly to the first edition of the task, we asked the participants to translate dialogues between two parties (i.e. *customer* and *agent*), where the *agent* writes in English and the *customer* writes in either German, French, or Brazilian Portuguese, depending on the language pair.

In order to evaluate the translation quality of the participating systems we used both automatic evaluation metrics and human judgement through MQM annotations. For the automatic evaluation metrics we used COMET (Rei et al., 2020), chrF (Popović, 2015), and SacreBLEU (Post, 2018), and for the human evaluation we used MQM (Lommel et al., 2014) performed with annotators specialized in explicit knowledge of translation errors and linguistics. Compared to the direct assessment evaluation (Graham et al., 2013, 2014, 2015) used in the previous edition, MQM annotations provide a more detailed analysis of the types and severities of the errors produced by the MT systems. MQM has also been shown to have a higher correlation with state-of-the-art metrics than direct assessments (Freitag et al., 2021).

This year, 18 submissions were received from 4 different teams, which have submitted outputs for both directions (i.e. *agent* and *customer*). Among these 4 teams, one team participated in all the three available language pairs, while the others focused only on en↔de. Details of their submissions and evaluation are described in §4 and §5.

2 The MAIA corpus

One of the biggest challenges of bilingual conversation translation, especially for Customer Support conversations, is the lack of appropriate publicly available datasets. To alleviate this issue, in the first edition of the Chat Translation Shared Task, Farajian et al. introduced the BConTrasT corpus that was based on a monolingual English corpus, Taskmaster-1 (Byrne et al., 2019). They translated the selected conversations into German mimicking a scenario in which an agent and a customer are communicating in their native languages. However, even this dataset was just an approximation of a real Customer Support conversation due to the fact that: 1) the original conversations in the Taskmaster-1 corpus were created by using crowdsourced workers interacting with each other to complete a specific task; and 2) the conversations were not truly genuine bilingual conversations since German segments were all just translations of the original English sentences.

This year, we made advancements by releasing the Unbabel's MAIA Dataset: a corpus that is truly composed of entire, genuine and original bilingual conversations from four different clients of the Unbabel database. The conversations are from clients that gave written consent on using this data for research purposes as long as in accordance with the General Data Protection Regulation (GDPR). In this corpus, the original segments of *customers* and *agents* are translated into their corresponding target languages via the MTPE process[1], done by the experienced translators of the Unbabel Community that have demonstrated consistently high quality within the respective language pair. MT segments were produced with a mixed of online MT services and internal ones. The corpus is released under the Creative Commons public license Attribution-NoDerivatives 4.0 International (CC BY-ND 4.0) and can be freely used for research purposes only. Please note that, as the license states, no commercial uses are permitted for this corpus. This data was collected within the MAIA Project (Martins et al., 2020).

The Unbabel's MAIA Dataset[2] contains more than 40k segments from more than 900 conversations in three language pairs (and a total of 6 language directions): English ↔ German (en↔de), English ↔ French (en↔fr) and English ↔ Portuguese (Brazil) (en↔pt-br). The breakdown of the corpus by language pair and direction is presented in Table 2. A sample conversation is presented in Table 1 and it shows that a conversation usually starts by the customer explaining the problem that led them to contact the *customer* support and is followed by the *agent* asking for more details in order to provide the necessary assistance.

Anonymization process. To make the conversations publicly available and in accordance with the General Data Protection Regulation (GDPR), we anonymized them first automatically by using the Unbabel proprietary anonymization tool and then by manually verifying the data. This resulted in 12 different anonymization categories, each presented by a specific token that are presented in Table 3. Importantly, Unbabel is also certified for ISO/IEC 27001:2013 Information Security Management Certification[3].

3 Task Description

Similarly to the first edition of the Chat Translation Shared Task, in this edition we focused on a critical challenge faced by international companies that are providing customer support in several different languages. One common approach to deal with this challenging requirement is centralizing the customer support with English speaking agents and having a translation layer in the middle to translate from the customer's language into the agent's language (e.g. English) and vice versa. The ideal solution for this environment needs to be able to properly handle all its aforementioned issues including the context-related challenges, the noisy inputs and multilingualism, among others.

In the second edition of the Chat Translation Shared Task we provide real genuine bilingual data for three different language pairs and encouraged the participants to make use of the bilingual context present in the conversations and to submit translations for both directions of the three language pairs. To emphasize on the importance of this aspect, we decided to rank the participating teams based on the overall quality of their primary submissions on both directions using a manual quality evaluation methodology through MQM annotations.

And, finally, we asked the participants to submit a maximum of three MT outputs per language pair direction, one primary and a maximum of two contrastive outputs. Due to time and budget constraints we performed the manual evaluation only for the primary submissions, while all the systems are evaluated using the automatic evaluation metrics (COMET, chrF, and SacreBLEU). For more details on the evaluation process please see §5.

3.1 Data

In the domain of customer support usually there is a very small amount of publicly available par-

[1] Machine Translation followed by a Post-Editing step.

[2] The full corpus can be downloaded from https://github.com/Unbabel/MAIA

[3] https://resources.unbabel.com/blog/unbabel-awarded-iso-iec-27001-2013-infor\mation-security-management-certification

MAIA corpus	en↔de	en↔fr	en↔pt-br
Number of conversations	496	264	164
Number of agent segments	8,509	9,911	4,741
Number of customer segments	9,468	5,115	3,674
Number of total (customer and agent) segments	17,977	15,026	8,415

Table 2: Number of conversations and segments in the MAIA corpus.

Token	Description
#NAME#	Person's names
#PRS_ORG#	Products, Services, and Organizations
#ADDRESS#	Address
#EMAIL#	E-mail address
#IP#	IP Address
#PASSWORD#	Password
#PHONENUMBER#	Phone number
#CREDITCARD#	Credit card number
#URL#	URL Address
#IBAN#	IBAN Address
#NUMBER#	Any number (all digits)
#ALPHANUMERIC_ID#	Any alphanumeric ID

Table 3: Anonymization tokens and their description.

allel data because of privacy and copyright issues that make releasing this kind of data difficult. In order to provide a more realistic setting, and due to the constraints outlined above, in this edition of the shared task, we provided participants with development and test sets only. The development sets can be divided into two types: SOURCE-ONLY, which that contains conversations without the human translations and PARALLEL, which that contains a smaller set of conversations with their corresponding human post-edited translations. The number of conversations and segments of the test and development sets per language pair and direction are presented in Table 4.

For training and validation purposes, participants were also allowed to use the training data of the general task (including the data of the previous editions), the data of the other tracks (eg. biomedical) and the other corpora (either parallel or monolingual) that are publicly available for research purposes including the data of the previous edition of the Chat Translation Task, BCONTRAST, as well as the corpora available on OPUS[4].

3.2 Baselines

In order to have a reasonable term of comparison for all the language direction, we used the large multilingual pre-trained M2M-100 model (Fan et al., 2021) with 418 million parameters. M2M is a multilingual MT model that supports all languages considered in this shared task. Moreover, since handling the context is one of the important challenges of the task we tried two baselines:

- A *sentence-level* baseline, where each utterance is passed separately to the model.
- A *context-level* baseline, where N consecutive utterances from the same conversation (and from the same direction) are passed and translated jointly by the model. In this paper we report the results of $N = 2$, that based on the automatic metrics performed the best on our validation sets.

While these models are not fine-tuned for *chat* conversation, they achieve good scores with automatic evaluation metrics and show the benefits of using context for this domain, even if they were not originally trained to use context.

We also report results of a larger version of the model (1B parameters) and different context sizes in Appendix A. In addition to these baselines, we also evaluated the results of four publicly available online MT systems on our test sets. In this paper we refer to them as Online-A, B, C, and D.

4 Participants

The participants were asked to submit at most three systems per language pair direction, one primary and a maximum of two contrastive ones. Moreover, the submitting team was required to explicitly indicate their primary and contrastive submissions. We received eighteen submissions from four different teams: BJTU-WeChat (two primaries and four contrastives), IITP-Flipkart (two primaries and four contrastives), HW-TSC (one primary and two contrastives), and Unbabel-IST (one primary and two contrastives). The first three teams participated only for en↔de, while the last team participated in all the language pairs and directions (i.e. en↔de, en↔fr, and en↔pt-br). Table 5 summarizes the participants and their affiliations.

[4] https://opus.nlpl.eu

	en↔de			en↔fr			en↔pt-br		
	source-only dev set	parallel dev set	parallel test set	source-only dev set	parallel dev set	parallel test set	source-only dev set	parallel dev set	parallel test set
Number of conversations	355	70	71	84	59	51	57	47	60
Number of total seg.	13,400	2,109	2,488	5,239	2,753	3,065	3,672	2,359	2,384
Number of agent seg.	6,389	1,006	1,113	3,305	1,750	1,937	2,007	1,353	1,381
Number of customer seg.	7,011	1,103	1,375	1,934	1,003	1,128	1,665	1,006	1,003

Table 4: Number of conversations and segments provided in the WMT 2022 Chat Translation Shared Task.

Team	Institution	Directions
BJTU-WeChat	Beijing Jiaotong University and WeChat	en↔de
HW-TSC	Huawei Translation Services Center	en↔de
IITP-Flipkart,	Indian Institute of Technology and Flipkart	en↔de
Unbabel-IST	Unbabel and Instituto Superior Técnico	en↔de, en↔fr, en↔pt-br
Online-A	A free publicly available online MT system	en↔de, en↔fr, en↔pt-br
Online-B	A free publicly available online MT system	en↔de, en↔fr, en↔pt-br
Online-C	A free publicly available online MT system	en↔de, en↔fr, en↔pt-br
Online-D	A free publicly available online MT system	en↔de, en↔fr, en↔pt-br

Table 5: The participating teams, their affiliations, and the directions that they participated.

All the participating systems follow a two step training in which a generic model is trained first on a large amount of publicly available data and then fine-tuned on the task data. The systems are different in the following aspects: i) the pre-training step, in which some use the publicly available models like mBART and Facebook-FAIR's WMT 2019, and the others train their own generic models, ii) the model architecture, in which some use deep encoder-decoder transformers, and others use multi-encoder transformers, iii) the fine-tuning stage and the data used in that step, and iv) the translation directions, in which some use bilingual models for each direction and some use a single multilingual model to cover all the language pairs and directions.

4.1 Systems

Here we briefly detail each participant's systems as described by the authors and refer the reader to the participant's submission for further details.

4.1.1 BJTU-WeChat

The joint submission of Beijing Jiaotong University and WeChat is an ensemble of deep Transformer models with 20 layers of encoder and 10 layers of decoder. Their models are firstly trained on the training corpora provided by the general track of WMT 2022. They are then fine-tuned on the training data of the chat translation track of WMT 2020 with several strategies to incorporate the potential context including the multi-encoder framework, speaker tag, and prompt-based fine-tuning.

Inspired by (Zhu et al., 2018) they proposed a Boosted Self-COMET-based Ensemble metric to evaluate the diversity of the generated hypotheses. As they report, it allows them to select some diverse, yet effective models from more than 100 models. Regarding the size of their models, the authors reports numbers that vary from 6.075 Billion to 6.881 Billion parameters.

4.1.2 IITP-Flipkart

IITP-Flipkart uses the Facebook-FAIR's WMT 2019 publicly available pre-trained models for en-de and de-en (Ng et al., 2019).[5] The models are based on the Transformer-big architecture (Vaswani et al., 2017). To fine-tune these models they follow a two-step procedure in which they first fine-tune the models on the training data of the Chat Translation track of WMT 2020 and then fine-tune the resulting models on the parallel validation set provided in the Chat Translation track of WMT 2022. To deal with the data scarcity issue of the task they combine the segments of agent and customer. To do so, for en-de, they use the agent subset of the above mentioned datasets as well as the customer segments by reversing their translation direction. The same applies to other direction.

[5] https://github.com/facebookresearch/fairseq/tree/main/examples/wmt19

For their primary submission they use a dual-encoder version of the WMT 2019 pre-trained FAIR model in which one encoder is used to encode the source context and the other one encodes the source segment. They use the weights of the encoder part of the pre-trained model to initialise the context-encoder weights. For the cross attention they use a weighted average of source-encoder and context-encoder attention. And for context they use the immediate previous source segment. Thus, the context can be either English or German, depending on the speaker of the previous utterance.

To analyze the impact of the context on the translation quality they experiment with a model that is trained with context and during inference it only uses the current sentence without any context. As they report, the results of this contrastive model confirm the observation of Li et al. that the improvement of the results are in some cases due to the fact that context acts as noise generator during training that makes the models more robust. And finally, their second contrastive model is a simple sentence level model that similarly as their primary model uses the Facebook-FAIR's WMT 2019 pre-trained model to initialise the weights. This model does not use any context.

As they reported, their primary submission is a model with 358 Million parameters. Their first contrastive model has the same number of parameters during training since it uses context, while during inference it uses only 312 million parameters since it does not use the context. This is the same number of parameters used by their second contrastive model that does not use any context at all.

4.1.3 HW-TSC

The Huawei Translation Services Center (HW-TSC)) team uses a deep transformer model with 25 layers of encoder and 6 layers of decoder. The model is pre-trained on the training data of the news track of WMT 2021. They used the bilingual validation set of the task to select in-domain data from the bilingual samples of the generic domain data. They reported the usage of self-training (i.e. forward translation), backward translation, model averaging, and context-aware translation.

4.1.4 Team Unbabel-IST

The joint submission of Unbabel and IST (Instituto Superior Técnico) uses the mBART50 model that has 12 layers of encoder and 12 layers of decoder. They fine-tuned the mBART50 model on a combination of the following two datasets: i) the in-domain parallel validation set, and ii) the samples similar to the validation set retrieved from the parallel generics corpus provided by the general track of WMT 2022. To find the similar samples they used LASER (Schwenk and Douze, 2017). At the inference time, their primary submission uses a retrieval-based approach in which for each segment of the test set the top-k nearest neighbors are retrieved from the following two data stores: i) the parallel in-domain validation set and ii) pool of the back-translated in-domain monolingual validation set of the task as well as the samples retrieved from the generic dataset that were used in the first stage of fine-tuning. Their first contrastive submission only uses the parallel validation set. Their second contrastive submission is the vanilla mBART50 model fine-tuned on the in-domain data, without the retrieval component.

Finally, concerning the model size, as they report it has the same number of parameters as mBART50, i.e. 761 million parameters.

5 Evaluation Procedures

Similarly to the previous edition, we evaluated the systems' performance both automatically and manually. This year we used COMET, chrF and SacreBLEU as the automatic metrics and MQM (Lommel et al., 2014) for the human evaluation. Due to time and budget constraints, the manual MQM evaluation was performed on the primary submissions only while all the submissions were evaluated using the automatic metrics. As mentioned earlier, the official rankings of the participating teams were based on the overall MQM score of their translations for the whole conversation, i.e. both *customer* and *agent* sides.

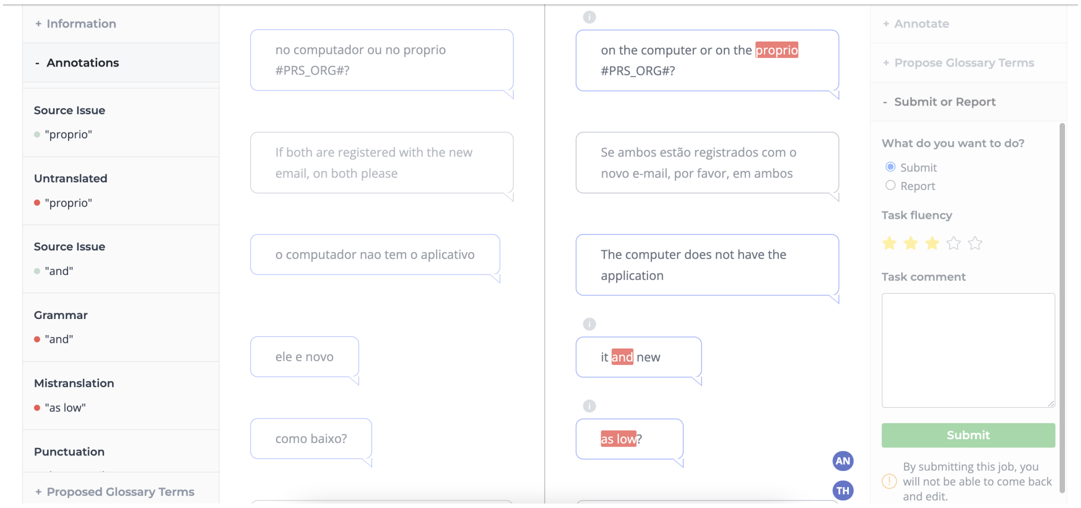
5.1 Automatic Evaluation

For the automatic evaluation of the systems' outputs we used COMET (`wmt20-comet-da`) (Rei et al., 2020), chrF (Popović, 2015), and SacreBLEU[6] (Post, 2018).

5.2 Human Evaluation

The human evaluation was performed by professional linguists and translators using an adaptation of the MQM framework (Lommel et al., 2014)

[6] We used version `2.1.0` with the signature `nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.1.0`



Source Issue
● "proprio"

Untranslated
● "proprio"

Source Issue
● "and"

Grammar
● "and"

Mistranslation
● "as low"

Punctuation

no computador ou no proprio #PRS_ORG#?

on the computer or on the proprio #PRS_ORG#?

If both are registered with the new email, on both please

Se ambos estão registrados com o novo e-mail, por favor, em ambos

o computador nao tem o aplicativo

The computer does not have the application

ele e novo

it and new

como baixo?

as low?

What do you want to do?
● Submit
○ Report

Task fluency
★★★☆☆

Task comment

By submitting this job, you will not be able to come back and edit.

Figure 1: The human evaluation was performed on the Unbabel's proprietary Annotation Tool by showing the annotations the whole conversation. The figure refers to an excerpt of an en↔pt-br conversation annotation.

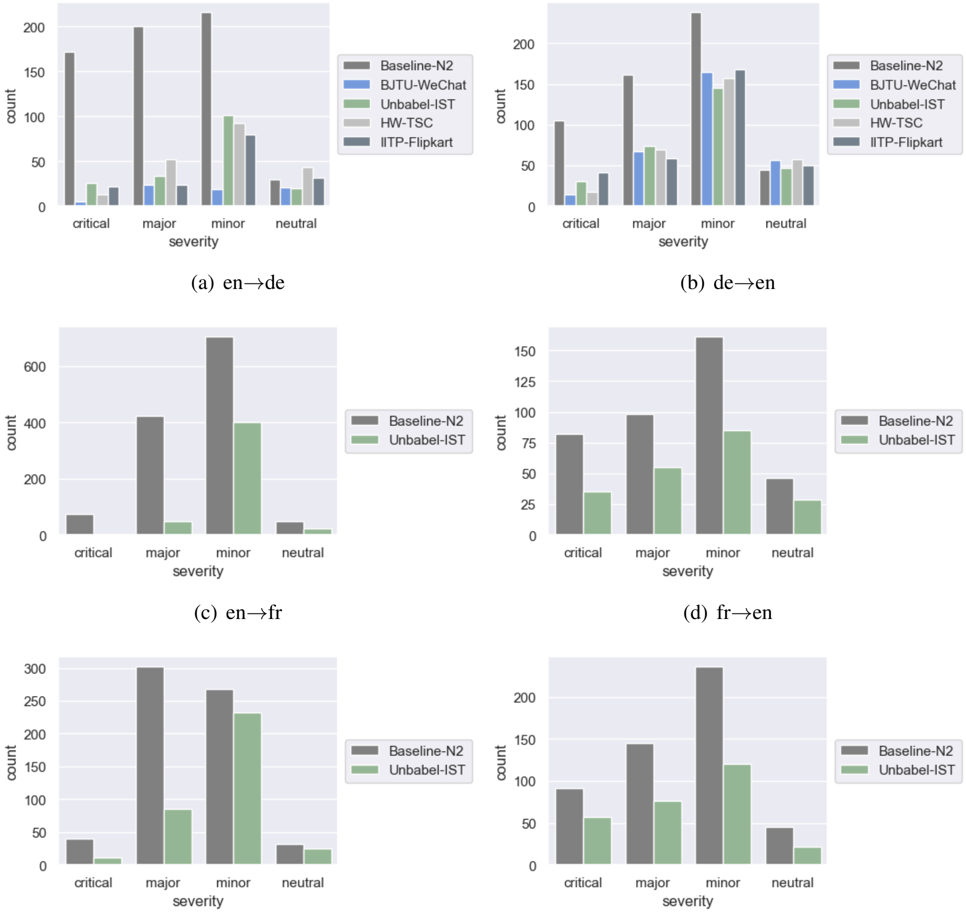
	en→de (agent)			de→en (customer)		
	COMET ↑	chrF ↑	BLEU ↑	COMET ↑	chrF ↑	BLEU ↑
Baselines						
Baseline without context	0.403	0.550	0.325	0.588	0.621	0.472
Baseline with context (N=2)	0.376	0.537	0.308	0.680	0.642	0.493
Primary submissions						
BJTU-WeChat	**0.810**	0.735	0.557	0.946	0.775	0.644
Unbabel-IST	0.774	0.733	0.557	0.915	0.737	0.612
IITP-Flipkart	0.768	0.730	0.549	0.907	0.729	0.582
HW-TSC	0.704	0.725	0.553	0.918	0.766	0.639
Contrastive submissions						
BJTU-WeChat, C1	0.804	0.731	0.551	0.948	0.780	0.646
BJTU-WeChat, C2	0.805	0.738	0.561	**0.951**	**0.778**	**0.648**
Unbabel-IST, C1	0.780	0.737	0.559	0.924	0.741	0.617
Unbabel-IST, C2	0.778	0.734	0.556	0.925	0.743	0.616
IITP-Flipkart, C1	0.769	0.730	0.550	0.905	0.729	0.582
IITP-Flipkart, C2	0.765	0.729	0.544	0.902	0.731	0.586
HW-TSC, C1	0.649	0.670	0.473	0.909	0.755	0.614
HW-TSC, C2	0.726	0.732	0.560	0.929	0.767	0.638
Online systems						
Online-A	0.758	**0.747**	**0.598**	0.903	0.733	0.579
Online-B	0.744	0.722	0.534	0.890	0.720	0.569
Online-C	0.717	0.707	0.515	0.877	0.730	0.575
Online-D	0.712	0.712	0.516	0.920	0.765	0.630

Table 6: Automatic metrics results for en↔de. The COMET scores are calculated with model `wmt20-comet-da`, and to calculate chrF and BLEU scores we used SacreBLEU.

that is tailored to assess Customer Support translated content (Gonçalves et al., 2022). The MQM-compliant typology used for this purpose is composed by:

- 8 parent categories, compliant with the newest version of the MQM framework[7]: *Accuracy*, *Linguistic Conventions*, *Terminology*, *Style*, *Locale Conventions*, *Audience Appropriateness*, *Design and Markup*, *Custom*;

[7] https://themqm.info/typology/



(a) en→de

(b) de→en

(c) en→fr

(d) fr→en

(e) en→ pt-br

(f) pt-br→en

Figure 2: Count of errors per severity for (a) en→de, (b) de→en, (c) en→fr, (d) fr→en, (e) en→pt-br, and (f) pt-br→en.

	en→fr (agent)			fr→en (customer)		
	COMET ↑	chrF ↑	BLEU ↑	COMET ↑	chrF ↑	BLEU ↑
Baselines						
Baseline without context	0.644	0.640	0.481	0.574	0.587	0.425
Baseline with context (N=2)	0.664	0.631	0.478	0.600	0.602	0.452
Primary						
Unbabel-IST	1.086	0.838	0.716	0.838	0.677	**0.544**
Contrastive						
Unbabel-IST, C1	1.082	0.836	0.712	0.840	0.676	0.542
Unbabel-IST, C2	**1.094**	**0.841**	0.718	**0.846**	0.675	0.542
Online systems						
Online-A	1.036	0.795	0.656	**0.846**	**0.678**	0.532
Online-B	1.085	0.838	**0.721**	0.827	0.669	0.517
Online-C	1.035	0.807	0.686	0.830	0.670	0.509
Online-D	1.044	0.788	0.618	0.819	0.673	0.521

Table 7: Automatic metrics results for en↔fr. The COMET scores are calculated with model `wmt20-comet-da`, and to calculate chrF and BLEU scores we used SacreBLEU.

	en→pt-br (agent)			pt-br→en (customer)		
	COMET ↑	chrF ↑	BLEU ↑	COMET ↑	chrF ↑	BLEU ↑
Baselines						
Baseline without context	0.824	0.681	0.495	0.610	0.631	0.471
Baseline with context (N=2)	0.863	0.675	0.493	0.675	0.653	0.496
Primary						
Unbabel-IST	1.077	0.771	0.621	0.849	0.689	0.547
Contrastive						
Unbabel-IST, C1	1.072	0.767	0.615	0.872	0.705	0.561
Unbabel-IST, C2	1.079	0.770	0.618	0.872	0.708	0.564
Online systems						
Online-A	0.965	0.725	0.551	**0.914**	**0.728**	**0.579**
Online-B	**1.084**	**0.791**	**0.647**	0.882	0.721	0.563
Online-C	1.069	**0.791**	0.643	0.887	0.726	0.559
Online-D	1.020	0.749	0.583	0.845	0.710	0.535

Table 8: Automatic metrics results for en↔pt-br. The COMET scores are calculated with model `wmt20-comet-da`, and to calculate chrF and BLEU scores we used SacreBLEU.

- 31 terminal nodes, including error types that are specific to MT, such as *MT Hallucination*[8] and customer support, such as *Source Issue*[9];
- 2 levels of granularity, composed by the 8 parent categories and the actual 31 terminal nodes (issue types) that annotators can use during the annotation process.

Regarding the severities attribution, we followed the same schema proposed in the MQM framework (Lommel et al., 2014), including a fourth severity, *Neutral*, to account for *Source Issue* errors. The definition of severities used in this evaluation are the following:

- *Neutral*: This severity degree is reserved only for the *Source Issue* category. This is used for linguistic issues that occur in the source text

[8] The *MT Hallucination* issue type is used when the MT generates a completely different translation that has no relation with the source text; the translation can still sound fluent and natural without reading the source, but the meaning is completely different. It is also used when the MT generates a chunk of repetitions in the target text or when the content is translated into gibberish: in other words, the machine generates an output made of non-words or repeated symbols.

[9] The *Source Issue* issue type needs to be used when there is an error in the target text and this is due to an issue in the source text. It can also be used when a part of the source text is written in the target language or in a different language, and the result is a mistranslation in the target.

	en↔de			en↔fr			en↔pt-br		
	agent	customer	overall	agent	customer	overall	agent	customer	overall
Baseline with context (N=2)	38.71	39.60	39.16	46.95	52.43	49.69	57.96	40.58	49.27
BJTU-WeChat	**96.44**	**80.09**	**88.27**	-	-		-	-	-
Huawei	88.33	79.02	83.68	-	-		-	-	-
Unbabel-IST	91.09	74.67	82.88	**90.08**	**77.21**	**83.65**	**84.16**	**69.01**	**76.59**
IITP-Flipkart	91.59	71.72	81.66	-	-		-	-	-

Table 9: MQM scores of the primary submissions of the participating teams, as well as the baseline MT systems.

that may have impact on the target translation and it is a signal of the overall quality of the source text to be translated;

- *Minor*: An error should be rated as minor if it does not lead to a loss of meaning and it does not confuse or mislead the user. It may, however, decrease the stylistic quality or fluency of the text, or make the content less appealing;
- *Major*: The usability or understandability of the content is impacted but it is still not unfit for purpose and the meaning of the content can be perceived as difficult to understand;
- *Critical*: The error severely changes the meaning of the original text. The reader cannot recover the actual meaning of the original text and the error carries health, safety, legal or financial implications to the end user/reader. In addition to this, a critical error also violates geopolitical usage guidelines, causes the application to crash or negatively modifies/misrepresents the functionality of the product or service. Finally, it can be offensive towards an individual or a group (a religion, race, gender, etc.).

To calculate the final MQM score per conversation the formula below is used:

$$\text{MQM} = 100 - \frac{I_{\text{Minor}} + 5 \times I_{\text{Major}} + 10 \times I_{\text{Crit.}}}{\text{Sentence Length} \times 100} \tag{1}$$

where I_{Minor} denotes the number of minor errors, I_{Major} the number of major errors and $I_{\text{Crit.}}$ the number of critical errors.

Figure 1 shows an excerpt of a pt-br customer conversation annotation, performed on a proprietary translation errors annotation tool from Unbabel. In this example, two *Source Issue* annotations are showcased, *proprio* and *and* that caused one Critical *Untranslated* error and one Critical *Grammar* error in the target text. In both cases, these examples outline some of the specificities of chat language and user-generated content, such as the lack of diacritics (Gonçalves et al., 2022) that can be observed in *proprio* and *e* on the source side (left pane) of the conversation. In the first case, *proprio*, a non-existent word in Portuguese, should have been written as *próprio*. As for the second case, *e* is a Portuguese conjunction that was translated literally into English, *and*, while the correct form should have been the verb *ser* (*to be* in English), conjugated in the 3rd person singular, *é*. The third error shown in Figure 1 shows yet another example of chat-specific language, such as the usage of a more idiomatic style (Gonçalves et al., 2022). The expression *como baixo?* refers to how *something can be downloaded from somewhere* and, besides its well-formedness in Portuguese, the style is idiomatic and conversational. The result is a *Mistranslation* error that refers to a literal translation into English. In this case, the meaning of the source text is completely lost and cannot be inferred by the reader.

Finally, as in the previous edition, we evaluated only a subsample of the full test set. For this, we randomly sampled conversations until the number of segments per direction was, at least, 500. We performed the annotations on both sides and calculated the overall conversation MQM score of each submission as the final score to use for the official ranking of the teams.

5.2.1 Customer Utility Analysis (CUA)

Besides reporting the overall MQM scores—the average MQM scores across conversations—, we decided to report, as a complement, a utility framework called Customer Utility Analysis (CUA) (Stewart et al., 2022). We decided to add this complementary analysis for two main reasons: 1) it gives us an idea of the quality across individual conversations; and 2) since MQM scores can be hardly understood without the context of a scale of reference or any direct connection the task-specific utility or value, CUA plots allow a better quality interpretation. This is possible because, as mentioned in §5.2, MQM is calculated by taking into account several factors, such as the total number of words of a conversation, the number of minor, major and critical errors and a severity multiplier. After the computation of the MQM scores at the conversation level, these are mapped into four different MQM buckets. In order to render this analysis more understandable from a visual point of view, we used a four color schema with the following MQM ranges:

- *Weak*: Dark Red (negative - 39 MQM)
- *Moderate*: Light Red (40 - 59 MQM)
- *Good*: Light Green (60 - 79 MQM)
- *Excellent*: Dark Green (80 - 100 MQM)

Ideally, we want the MT systems to have larger green and smaller red buckets, indicating less errors in the MT outputs and higher MQM scores.

6 Discussion

In this section we analyze the results of the automatic and human evaluation of the systems from different aspects.

6.1 Official ranking of the systems

The MQM scores of all the primary submissions as well the baselines (with context size 2) are presented in Table 9. As can be observed, in addition to the MQM score of each direction, the overall conversation-level MQM scores are also reported for each system.

Based on the overall MQM scores, the BJTU-WeChat team ranks first for the en↔de language pair, achieving higher MQM scores for both directions. This is consistent with the automatic scores of the systems reported in Table 6. BJTU-WeChat is followed by Huawei, Unbabel-IST, and IITP-Flipkart. As we can see in the distribution of the error severities in Figure 2, BJTU-WeChat produces significantly less critical and major errors in both directions. In the Neutral category we can see that all the systems perform almost the same, including the baselines. Based on this observation and the definition of this severity category (§5.2) we can infer that all the systems handle source-related issues more or less similarly. This calls for methods that are more reliable to source sentence issues, in particular for the *customer* side in which we have a significantly larger amount of issues when compared to the *agent* side.

By looking at the distribution of the error types presented in Table 15 we can see that "Mistranslation" is the most frequent error for all the systems. Given the definition of this error[10] and the fact that there was no in-domain training data for the given domain it was expected to see a large number of these errors in the outputs of all the MT systems.

For the en↔de language pair we received submissions from four teams, however, for en↔fr and en↔pt-br we received the outputs of one participating team only, making it more difficult to do an in-depth analysis on the results. The MQM scores of the baselines and the participating team are reported in Table 9, and their automatic scores can be found in Tables 7 and 8, respectively. As we can see, the Unbabel-IST systems outperform the baselines significantly both in terms of the manual MQM scores as well as the automatic metrics.

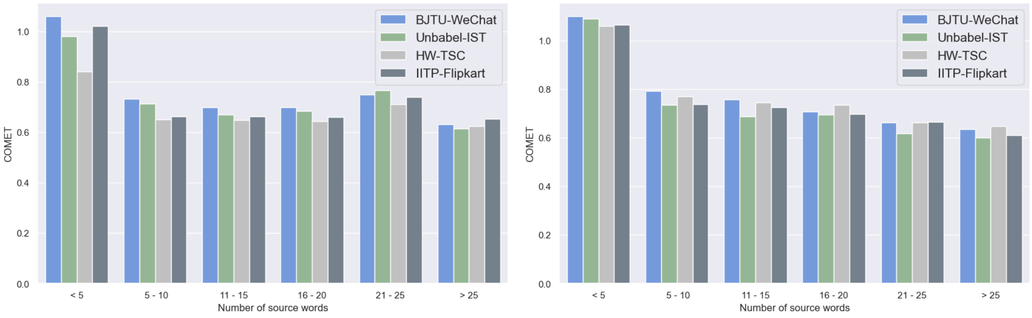
6.2 Computational efficiency of the approaches

The results of the primary submissions and the online systems (Table 6) shows that there is a big difference between the BJTU-WeChat submission and the other systems. As reported by the participants, this system is the only submission that uses an ensemble of a large number of models that makes it the least computationally efficient solution for the problem. The other submissions obtain results similar or better than the online systems and do not resort to model ensembling, making them more computationally efficient than the winning submission. The applicability and the computational efficiency of the models is one of the factors that we plan to pay more attention to in the future editions of the shared task.

6.3 Noisy source and its impact on MT quality

By comparing the MQM scores of the two directions (i.e. *agent* and *customer*) we can see that independently of the language pair, the scores of the *customer* side are significantly lower than the

[10] The word or phrase being translated wrongly according to the domain of interest.



(a) en→de

(b) de→en

Figure 3: COMET scores of the segments in different buckets based on the number of words in the source for (a) *agent* direction (en→de) and (b) *customer* direction (de→en).

scores of the *agent* side. This is in contrast to the previous observations that translating into English is usually easier than translating from English (Akhbardeh et al., 2021). We assume this is partially due to the different amounts of noise in the source segments of each direction and their impact on the final quality of the MT systems. To support our claim, we analyzed the source side of all the text conversations with a proprietary rule-based tool developed at Unbabel to detect spelling and grammatical errors, perform writing style checks (related to the formality of the text), among the detection of other types of issues that are specific to the content type of customer service conversations. As we can see in Table 10 the *customer* segments contain a larger degree of noise, up to 4 times, with respect to the *agent* side. We then proceeded by splitting the source segments of each direction into two sets of *noisy* and *non-noisy* categories and analyzed the quality of the models on each set separately. As we can see in Figure 5 the quality of the models on the noisy samples is significantly lower compared to the non-noisy samples. This is in line with the findings of Gonçalves et al., in which customers requiring customer support help usually exhibit high levels of impatience and frustration, that might be translated into agrammatical and unstructured text with lexical choices that often result in a degradation of the machine translation output.

6.4 Sentence length and MT quality

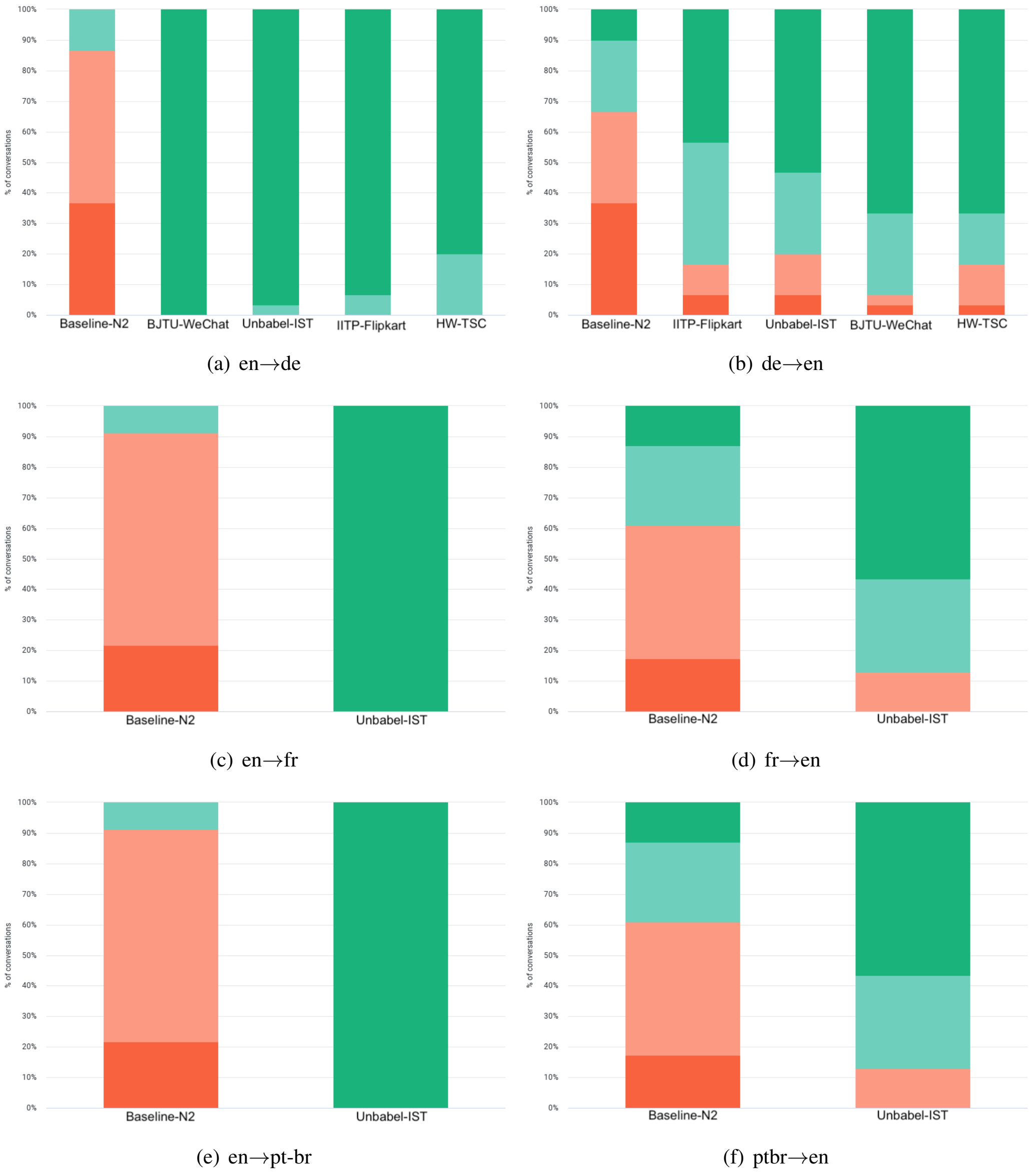
Looking at the test sets we can see varying lengths of source sentences, with the majority of them being very short segments (see Figure 6). To understand the impact of the sentence lengths on the final quality of the MT systems we grouped the input sentences into six buckets and measured the COMET score of each bucket (see Figure 3 for the COMET scores of the primary submissions of en→de and de→en). Independently of the direction, we can see that: 1) systems perform fairly similarly within each bucket; and 2) systems' performances tend to decrease as the number of source words increases. The pattern is very similar for the other language pairs and directions.

	agent	customer
en↔de	55	105
en↔fr	40	116
en↔pt-br	24	95

Table 10: The number of noisy source segments in each side of the test conversations.

6.5 MT systems and CUA analysis

As mentioned in §5.2.1, CUA analysis provides complementary information to have a more clear understanding on the distribution of MQM scores. The bucketing approach used in CUA helps to easily interpret the quality of the MT systems. By looking into the de→en primary submissions, we can see that the BJTU-WeChat system not only outperforms the other systems significantly, but also produces the highest number of *excellent* translations. We can also see that, in general, the *agent* directions are easier to translate. In fact, no system produces any *Weak* or *Moderate* translations for this direction, while we can see a large number of *Weak* or *Moderate* ones in the outputs of all the systems for the *customer* direction.



(a) en→de

(b) de→en

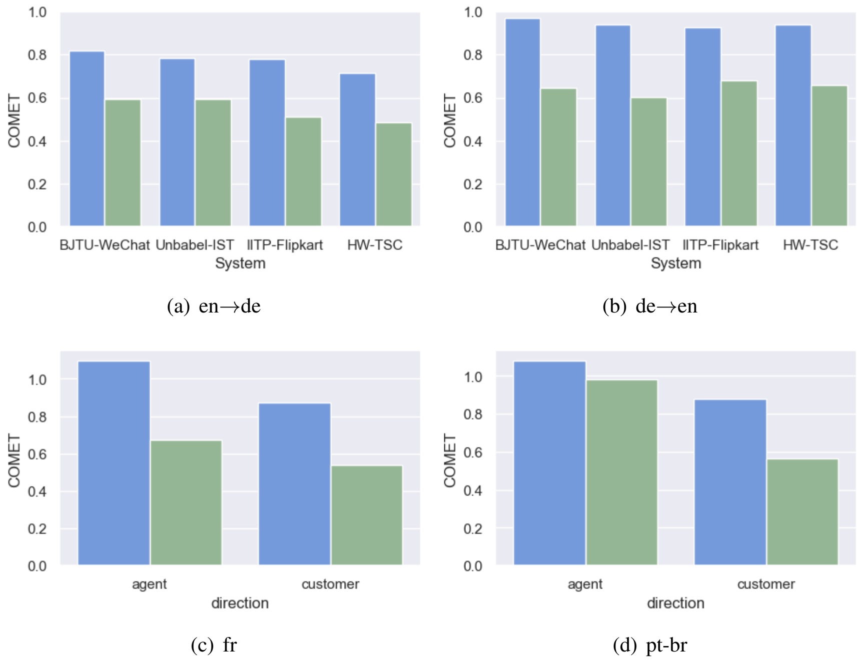
(c) en→fr

(d) fr→en

(e) en→pt-br

(f) ptbr→en

Figure 4: CUA plots for (a) en→de, (b) de→en, (c) en→fr, (d) fr→en, (e) en→ptbr, and (f) ptbr→en. Color schema: dark red (weak), light red (moderate), light green (good), and dark green (excellent).



(a) en→de

(b) de→en

(c) fr

(d) pt-br

Figure 5: COMET scores of the primary submissions on the *noisy* (in green) and *non-noisy* samples (in blue) of the test sets. The noises were detected by a proprietary tool developed at Unbabel. (a) shows the results on the agent direction (en→de), (b) shows the results on the customer direction (de→en), while (c) and (d) show the results of the only primary submissions (i.e. Unbabel-IST) for en↔fr and en↔pt-br, respectively.

6.6 Usage of context

All the four participating teams reported the incorporation of context in their experiments. But, depending on the approach, and the data they used as the context, they obtained different results and draw different conclusions. BJTU-WeChat used a simple prompt learning approach in which they add two preceding bilingual contexts at the tail of each utterance with a special token to indicate the boundary of the context. Their results show slight performance gains over the models that do not use the context. HW-TSC explores a similar approach but no promising results can be observed. This can be due to different factors like implementation details, the size and the combination of the data used as context, among other factors. For more details about the approaches and their difference please refer to their system description papers.

Differently than the BJTU-WeChat and the HW-TSC teams that use variations of the concatenation approach, IITP-Flipkart reports using an additional context encoder for incorporating context information. However, based on the test sets results we cannot observe any meaningful improvement over the system that does not incorporate the context, at least with automatic evaluation metrics.

Finally, Unbabel-IST report that in the few experiments they performed using context they did not observe any meaningful improvement on the performance of their models.

7 Conclusions

We presented the results of the WMT 2022 Chat Translation Shared Task. This year, we provided the participants with anonymized genuine bilingual Customer Support conversations for development and test sets. The conversations are part of the MAIA corpus, a corpus that we introduced here for the first time that aim to provide the best possible research ground for this very particular domain.

Four different teams participated in en↔de and one team participated also for en↔fr, and en↔pt-br. All participants covered both directions (i.e. *customer* and *agent*). We evaluated submissions with automatic metrics (i.e. COMET, chrF, and SacreBLEU) and primary submissions with MQM human evaluation. The MQM evaluations were conducted under an adaptation of the

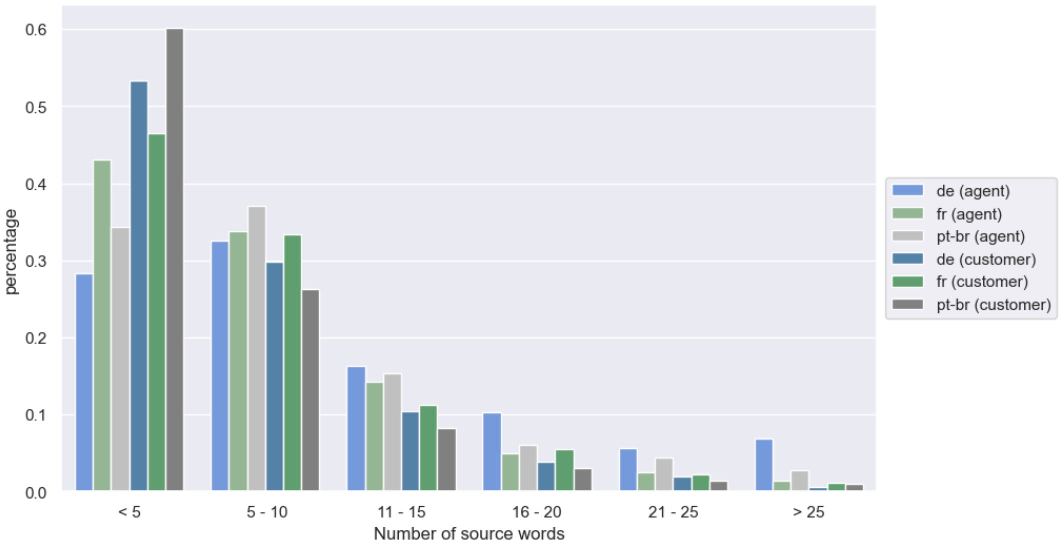


Figure 6: Percentage of segments when bucked according to the number of source words per lp and direction.

MQM framework (Lommel et al., 2014), that is tailored to assess Customer Support translated content (Gonçalves et al., 2022), providing a rich analysis of the type of errors that, we hope, will foster future MT research in this domain.

Acknowledgements

We would like to thank Vera Cabarrão and Miguel Menezes for their work in the anonymization of the data; Tânia Vaz, Madalena Gonçalves, and Maximilian Kohl for their help conducting the MQM annotations; Ricardo Rei and João Alves for useful discussions regarding the methodology used; and Pedro Gonçalves for the legal advise that made the release of the data possible. This work was supported by the P2020 programs MAIA (contract 045909), by the European Research Council (ERC StG DeepSPIN 758969), by the Fundação para a Ciência e Tecnologia through contract UIDB/50008/2020 and by the European Union's Horizon Europe Research and Innovation Actions under Grant Agreement 101070631 (UTTER).

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4506–4517.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Çelebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22:107:1–107:48.

M. Amin Farajian, António V. Lopes, André F. T. Martins, Sameen Maruf, and Gholamreza Haffari. 2020. Findings of the WMT 2020 shared task on chat translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 65–75, Online. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021.

Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Madalena Gonçalves, Marianna Buchicchio, Craig Stewart, Helena Moniz, and Alon Lavie. 2022. Agent and user-generated content and its impact on customer support MT. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 201–210, Ghent, Belgium. European Association for Machine Translation.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2014. Is machine translation getting better over time? In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 443–451, Gothenburg, Sweden. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2015. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, 23:3 – 30.

Bei Li, Hui Liu, Ziyang Wang, Yufan Jiang, Tong Xiao, Jingbo Zhu, Tongran Liu, and Changliang Li. 2020. Does multi-encoder help? a case study on context-aware neural machine translation. In *proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Arle Lommel, Hans Uszkoreit, and Aljoscha Burchardt. 2014. Multidimensional quality metrics (mqm): A framework for declaring and describing translation quality metrics. *Revista Tradumàtica: tecnologies de la traducció*, (12):455–463.

André F. T. Martins, Joao Graca, Paulo Dimas, Helena Moniz, and Graham Neubig. 2020. Project MAIA: Multilingual AI agent assistant. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 495–496, Lisboa, Portugal. European Association for Machine Translation.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*, pages 157–167. Association for Computational Linguistics.

Craig A Stewart, Madalena Gonçalves, Marianna Buchicchio, and Alon Lavie. 2022. Business critical errors: A framework for adaptive quality feedback. In *Proceedings of the 15th Biennial Conference of the Association for Machine Translation in the Americas (Volume 2: Users and Providers Track and Government Track)*, pages 231–256, Orlando, USA. Association for Machine Translation in the Americas.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st Conference on Neural Information Processing Systems*, volume 30.

Tao Wang, Chengqi Zhao, Mingxuan Wang, Lei Li, and Deyi Xiong. 2021. Autocorrect in the process of translation — multi-task learning improves dialogue machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 105–112, Online. Association for Computational Linguistics.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*.

A Baseline Results with Other Context Sizes

Table 11: M2M-418M results on the *development* set with various context sizes

Lang	Direction	Context Size	BLEU	ChrF	COMET
de	agent	all	31.94	53.25	0.2504
		0	**35.24**	**57.17**	**0.4168**
		1	33.80	56.05	0.3910
		2	33.89	55.96	0.3811
		3	33.40	55.76	0.3648
		5	33.07	55.23	0.3493
	customer	all	47.14	62.05	0.6114
		0	45.98	60.81	0.5426
		1	**48.28**	**62.80**	**0.6326**
		2	47.11	62.06	0.6163
		3	47.23	62.08	0.6073
		5	47.52	62.41	0.6225
fr	agent	all	45.67	61.02	0.5105
		0	54.14	69.47	0.7984
		1	**54.72**	69.83	**0.8173**
		2	53.58	68.81	0.7978
		3	53.68	69.00	0.7973
		5	52.69	68.00	0.7750
	customer	all	48.14	63.77	0.6784
		0	46.51	62.29	0.6382
		1	**48.35**	63.53	0.6526
		2	48.05	**63.61**	**0.6834**
		3	48.52	64.06	0.6786
		5	48.17	63.74	0.6753
pt-br	agent	all	45.60	63.25	0.7801
		0	**50.38**	**68.84**	0.8645
		1	49.67	67.95	**0.9129**
		2	49.94	67.95	0.9029
		3	49.11	67.41	0.9116
		5	48.67	66.95	0.8935
	customer	all	47.10	62.29	0.6449
		0	44.71	59.95	0.5851
		1	46.88	62.06	0.6332
		2	**47.24**	**62.31**	0.6437
		3	46.96	**62.31**	**0.6491**
		5	47.30	62.53	0.6514

Table 12: M2M-1B results on the *development* set with various context sizes

Lang	Direction	Context Size	BLEU	ChrF	COMET
de	agent	all	33.64	50.81	0.0070
		0	**43.36**	63.90	**0.4696**
		1	39.86	59.56	0.2938
		2	36.96	54.70	0.1669
		3	35.04	52.66	0.0926
		5	34.52	51.63	0.0505
	customer	all	49.84	64.41	0.5192
		0	**60.20**	74.03	**0.8307**
		1	59.44	72.44	0.7976
		2	57.08	70.71	0.7620
		3	57.87	71.52	0.7889
		5	57.18	70.73	0.7554
fr	agent	all	48.67	65.78	0.7100
		0	**55.16**	72.33	0.8718
		1	52.67	70.21	0.8857
		2	51.75	69.47	0.8873
		3	52.58	70.45	**0.8988**
		5	49.89	68.94	0.8122
	customer	all	50.02	64.56	0.6510
		0	50.33	64.73	0.6434
		1	50.18	64.79	**0.6626**
		2	**50.57**	65.21	0.6550
		3	50.24	64.86	0.6546
		5	50.31	64.86	0.6551
pt-br	agent	all	48.63	64.10	0.6409
		0	49.26	64.31	**0.6884**
		1	49.51	64.46	0.6362
		2	49.76	64.92	0.6449
		3	**49.79**	65.21	0.6597
		5	48.56	64.03	0.6422
	customer	all	48.48	63.51	0.6401
		0	45.99	61.18	**0.6427**
		1	**48.98**	63.89	0.6424
		2	48.22	62.57	0.6167
		3	48.30	63.17	0.6415
		5	48.25	63.13	0.6241

	en→fr (agent)		fr→en (customer)	
	Baseline-N2	Unbabel-IST	Baseline-N2	Unbabel-IST
Addition	39	17	37	12
Agreement	6	6	2	3
Capitalization	55	32	12	0
Currency Format	0	2	0	0
Date/Time Format	2	2	4	2
Grammar	90	24	36	22
MT Halucination	8	0	19	0
Mistranslation	469	60	113	83
Omission	21	11	42	28
Punctuation	170	98	18	4
Register	2	0	0	0
Source Issue	50	22	46	29
Spelling	19	20	2	0
Unnatural Flow	3	1	0	0
Untranslated	207	2	25	7
Whitespace	86	161	2	0
Word Order	26	18	16	4
Wrong Named Entity	0	0	13	10

Table 13: Counts per error type for fr .

	en→pt-br (agent)		pt-br→en (customer)	
	Baseline-N2	Unbabel-IST	Baseline-N2	Unbabel-IST
Addition	14	8	47	13
Agreement	9	1	0	1
Capitalization	24	18	27	19
Currency Format	3	2	0	0
Grammar	53	31	54	28
MT Halucination	1	0	28	6
Mistranslation	224	99	145	100
Omission	25	30	84	42
Punctuation	131	109	19	15
Register	2	0	0	0
Source Issue	32	25	46	23
Spelling	2	2	0	0
Unnatural Flow	8	2	0	0
Untranslated	97	5	22	4
Whitespace	4	5	8	4
Word Order	10	4	25	15
Wrong Language Variety	1	6	0	0
Wrong Named Entity	2	6	12	5

Table 14: Counts per error type for pt-br.

	en→de (agent)					de→en (customer)				
	Baseline-N2	BJTU-WeChat	Unbabel-IST	IITP-Flipkart	HW-TSC	Baseline-N2	BJTU-WeChat	Unbabel-IST	IITP-Flipkart	HW-TSC
Addition	19	1	7	4	7	38	10	23	14	19
Agreement	1	3	0	1	0	0	0	1	0	0
Capitalization	2	1	2	1	1	31	7	4	6	5
Currency Format	0	0	0	0	0	0	1	0	0	0
Date/Time Format	0	0	0	0	0	2	1	2	2	4
Grammar	82	1	35	13	22	46	25	32	30	23
Inconsistency	0	0	1	0	0	0	2	0	2	1
MT Halucination	3	0	0	0	1	18	0	4	0	0
Measurement Format	0	0	0	0	0	1	1	1	1	1
Mistranslation	258	34	58	51	81	179	66	84	56	72
Omission	81	2	11	11	10	87	77	60	85	78
Punctuation	6	0	4	10	10	28	19	9	17	15
Register	18	0	0	0	3	0	0	0	0	0
Source Issue	31	22	20	32	43	47	57	47	50	58
Spelling	0	3	0	1	2	1	0	0	1	0
Untranslated	81	0	19	15	6	15	9	3	5	4
Whitespace	7	0	14	14	4	12	7	5	30	3
Word Order	29	0	8	5	10	42	17	18	13	17
Wrong Named Entity	0	2	2	0	0	2	3	3	2	2

Table 15: Counts per error type for de.

Findings of the First WMT Shared Task on Sign Language Translation (WMT-SLT22)

Mathias Müller University of Zurich
Sarah Ebling University of Zurich
Eleftherios Avramidis DFKI Berlin
Alessia Battisti University of Zurich
Michèle Berger HfH Zurich
Richard Bowden University of Surrey
Annelies Braffort University of Paris-Saclay
Necati Cihan Camgöz Meta Reality Labs
Cristina España-Bonet DFKI Saarbrücken
Roman Grundkiewicz Microsoft
Zifan Jiang University of Zurich
Oscar Koller Microsoft
Amit Moryossef Bar-Ilan University
Regula Perrollaz HfH Zurich
Sabine Reinhard HfH Zurich
Annette Rios University of Zurich
Dimitar Shterionov Tilburg University
Sandra Sidler-Miserez HfH Zurich
Katja Tissi HfH Zurich
Davy Van Landuyt European Union of the Deaf

Abstract

This paper presents the results of the First WMT Shared Task on Sign Language Translation (WMT-SLT22)[1]. This shared task is concerned with automatic translation between signed and spoken[2] languages. The task is novel in the sense that it requires processing visual information (such as video frames or human pose estimation) beyond the well-known paradigm of text-to-text machine translation (MT). The task featured two tracks, translating from Swiss German Sign Language (DSGS) to German and vice versa. Seven teams participated in this first edition of the task, all submitting to the DSGS-to-German track. Besides a system ranking and system papers describing state-of-the-art techniques, this shared task makes the following scientific contributions: novel corpora, reproducible baseline systems and new protocols and software for human evaluation. Finally, the task also resulted in the first publicly available set of system outputs and human evaluation scores for sign language translation.

1 Introduction

This paper presents the outcome of the First WMT Shared Task on Sign Language Translation (WMT-SLT22). The focus of this shared task is automatic translation between signed and spoken languages. Recently, Yin et al. (2021) called for including signed languages in NLP research. We regard our shared task as a direct answer to this call. While WMT has a long history of shared tasks for spoken languages (Akhbardeh et al., 2021), this is the first time that signed languages are included in a WMT shared task.

Sign language translation requires processing visual information (such as video frames or human pose estimation) beyond the well-known paradigm of text-to-text machine translation (MT). As a consequence, solutions need to consider a combination of Natural Language Processing (NLP) and computer vision (CV) techniques.

In the field of sign language MT there is a general lack of suitable and freely available datasets and code. For this reason it was necessary for us to build and distribute novel training corpora and we also published reproducible baseline code. Likewise, existing protocols and toolkits for human evaluation had to be adapted to support sign languages.

In this first edition of the shared task we considered one language pair: Swiss German Sign Language (DSGS) and German. We offered two tracks: DSGS-to-German translation and German-to-DSGS translation.

Seven teams participated in the task, which we consider a success. All teams submitted to the DSGS-to-German track, while there were no submissions to the German-to-DSGS track.

The remainder of this paper is organized as follows:

[1] https://www.wmt-slt.com/

[2] In this paper we use the word "spoken" to refer to any language that is not signed, no matter whether it is represented as text or audio, and no matter whether the discourse is formal (e.g. writing) or informal (e.g. dialogue).

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 744 - 772
December 7-8, 2022 ©2022 Association for Computational Linguistics

- We give some background on sign languages and sign language processing in §2.
- We describe the shared task tracks and submission procedure in §3.
- We report on the corpora we built and distributed specifically for this task in §4 and §5.
- We describe all submitted systems, including our baseline in §6.
- We ran both an automatic and a human evaluation. We explain our evaluation in §7.
- We share the main outcomes in §8 and discuss in §9.

2 Background

We consider sign language processing (SLP) a sub-area of Natural Language Processing (NLP), and automatic sign language translation (SLT), a more narrowly focused discipline within SLP.

We first give an introduction to sign languages (§2.1) and describe the societal and academic relevance of SLP (§2.2). Then we give an overview of SLP in general (§2.3), of SLT in particular (§2.4) and finally motivate this shared task (§2.5).

2.1 Sign languages

Sign languages (SLs) are the natural languages used in deaf communities. Contrary to the popular belief that sign language is universal, hundreds of different SLs have been documented so far. They are still scarcely described and under-resourced. For example, few reference grammars exist, lexicons only have partial coverage and existing corpora are small.

Nature of sign languages Sign languages are visuo-gestural languages. A person expresses themselves using many parts of the body (hands and arms, but also face, mouthing, gaze, shoulders, torso, etc.) while the interlocutor perceives the message through the visual channel. The linguistic system of SLs makes use of these specific linguistic cues. Information is expressed simultaneously (as opposed to the sequential nature of spoken language), organized in three-dimensional space, and iconicity plays a central role (Woll, 2013; Perniss et al., 2015; Slonimska et al., 2021).

Writing systems To date, SLs do not have a written form or graphical system for transcription that is universally accepted (Pizzuto and Pietrandrea, 2001; Filhol, 2020). Several notation systems, such as HamNoSys (Hanke, 2004) or SignWriting (Sutton, 1990; Bianchini and Borgia, 2012), are used in research or teaching but are rarely adopted as a writing system in everyday life.

A common misconception among MT researchers is that transcribed glosses are a full-flegded writing system for sign languages. In reality, glossing is a linguistic tool, useful for annotating corpora for linguistic studies (Johnston, 2010). Glosses are not a means of writing SL, and they do not adequately represent the meaning of an SL utterance. Importantly, “deaf people do not read or write glosses” in everyday life (Anonymous, 2022). Moreover, glosses mostly consist of words taken from the surrounding spoken language, which is generally only a second language to deaf signers (§2.2, societal relevance).

2.2 Relevance of sign language processing

SLP is a research area with high potential impact, as it is relevant in a societal and academic sense.

Societal relevance The overall aim of SLP is to provide language technology for sign languages, which currently are somewhat overlooked. The vast majority of NLP systems are designed for spoken languages, not for signed languages. This means that more research in SLP could result in more equal access to language technology.

The more specific goal of SLT is to facilitate communication between deaf and hearing communities. There is a need for this because speakers of spoken languages and signers of sign languages experience communication difficulties (the same kind of difficulties encountered by speakers of different spoken languages). We emphasize that deaf and hearing people could benefit from such technologies in equal measure.[3]

Besides aiding direct communication, SLT would improve accessibility to spoken language content, given that spoken languages are often a second language for deaf people, where they exhibit varying proficiency. The reverse direction can also be useful, for example to automatically

[3] We distance ourselves from the harmful view that only deaf people are in need (of access to spoken language discourse). Language barriers are inherently two-way, and addressing them involves both parties.

subtitle signed content to make it accessible to people who do not know SLs (Bragg et al., 2019).

Academic relevance In the field of Natural Language Processing (NLP), working on SLs is highly innovative and timely. Recently, a call for more inclusion of signed languages in NLP (Yin et al., 2021) was widely publicized, and an ACL initiative for Diversity and Inclusion[4] targets SL processing as well.

2.3 Sign language processing

Sign language processing is an interdisciplinary field, bringing together research on NLP and computer vision, among other disciplines (Bragg et al., 2019). For a general overview in the context of NLP see Yin et al. (2021); Moryossef and Goldberg (2021).

Tasks SLP involves a variety of (sub)tasks with individual challenges. Widely known tasks are sign language recognition, sign language translation and sign language production (or *synthesis*). Sign language recognition usually refers to identifying individual signs from videos, see Koller (2020) for an overview. Sign language translation refers to systems that transform sign language data to a second language, no matter whether signed or spoken, see De Coster et al. (2022) for a comprehensive survey. Finally, sign language production refers to rendering sign language as a video, using methods such as avatar animation (Wolfe et al., 2022) or video generation.

SLP research is challenging for a number of different reasons. The ones we chose to highlight here are linguistic properties, availability of data and availability of basic NLP tools.

Linguistic challenges SLP is challenging because of the characteristics of sign languages (§2.1), for instance multilinearity, use of the signing space and iconicity. As explained earlier, SLP needs to take into account manual and non-manual cues in order to capture a complete linguistic picture of an SL utterance (Crasborn, 2006). Information is presented simultaneously, rather than sequentially. Signing makes frequent use of indexing strategies for example to identify referents introduced earlier in the discourse or timelines (Engberg-Pedersen, 1993).

Sign languages have an established vocabulary but are also lexically productive to allow for definition of new signs or constructions to be used to depict entities or situations (Johnston, 2011).

Availability of data Given the current research landscape in NLP, sign languages are under-resourced. An analysis by Joshi et al. (2020) places all sign languages considered in this study in the category "left behind" (together with many spoken languages). Existing resources are small and also heterogeneous. They are created under a variety of circumstances and vary in quality (e.g. video resolution), signer demographics (e.g. deaf vs. hearing signers), richness of annotation (e.g. glosses, sentence segmentation, translation to a spoken language) and linguistic domain (e.g. only weather reports).

Also, not all corpora are easily accessible online and some have restrictive licenses that disallow NLP research. A survey of SL corpora available in Europe can be found in Kopf et al. (2021).

Lack of basic linguistic tools SLP currently lacks fundamental NLP tools that are readily available for spoken languages. Such tools include automatic language identification (Monteiro et al., 2016), sign segmentation (De Sisto et al., 2021), sentence segmentation (Ormel and Crasborn, 2012; Bull et al., 2020) and sentence alignment (Varol et al., 2021). Although there are experimental solutions, they are not yet viable in practice.

Tools like these would be crucial to create better corpora by constructing them automatically, as is routinely done for spoken languages (Bañón et al., 2020), and develop better high-level NLP solutions.

2.4 Sign language translation

In recent years, different methods to tackle SLT have been proposed, most of them suggesting a cascaded system where a signed video is first converted to an intermediate representation and then to spoken text (similarly for text-to-video translation). Intermediate representations (with individual strengths and weaknesses) include pose estimation (§5.3), glosses or writing systems such as HamNoSys (§2.1, writing systems).

There is existing work on gloss-to-text translation and vice versa (e.g. Camgöz et al. 2018; Yin and Read 2020), pose-to-text translation and

[4] https://www.2022.aclweb.org/dispecialinitiative

vice versa (e.g. Ko et al. 2019; Saunders et al. 2020a,b,c) and systems involving HamNoSys (e.g. Morrissey 2011; Walsh et al. 2022). Recently, direct video-to-text translation was also proposed by Camgöz et al. (2020a,b). For rendering sign language output, avatars are commonly used (Wolfe et al., 2022), as well as methods to generate videos of realistic signers (e.g. Saunders et al. 2022).

Parallel datasets In terms of datasets, past work in SLT can be characterized as focusing very much on a narrow linguistic domain, most of the work was done on one single data set called `RWTH-PHOENIX Weather 2014T` (Forster et al., 2014). PHOENIX has a size of 8k sentence pairs and contains only weather reports. The biggest parallel sign language corpus to date, the Public DGS Corpus (Hanke et al., 2020), contains roughly 70k sentence pairs.

Thus, there is a clear shortage of usable parallel corpora and existing ones are orders of magnitude smaller than what is considered an acceptable size for spoken language MT (as a rule of thumb, at least hundreds of thousands of sentence pairs). Nevertheless, there are plenty of spoken languages that also have little parallel data and MT methods have been developed specifically for low-resource MT (Sennrich and Zhang, 2019).

Evaluation For spoken language MT a variety of automatic metrics exist. These include more conventional, string-based metrics such as BLEU (Papineni et al., 2002) or chrF (Popović, 2015), as well as recent, learned metrics based on embeddings like COMET (Rei et al., 2020). In the context of SLT, no automatic metrics are validated empirically, but if the target language is spoken, many existing metrics are reasonable to use. However, if sign language is the target language, no automatic metric is known at the time of writing and the only viable evaluation method is human evaluation. A human evaluation of SLT systems has never been conducted on a large scale before, and there are open questions regarding the exact evaluation methodology and what the ideal profile (e.g. hearing status, language proficiency) for evaluators should be.

2.5 Motivation for this shared task

Our main motivation is that sign languages are natural languages (§2.1) that are currently overlooked in NLP and SLT research (§2.3, §2.4). The shared task brings this topic to the attention of MT researchers. We decided to create a new shared task as opposed to other activities since we believe this format has a unique potential to foster progress in MT and to also make progress measurable over time.

Concrete ways in which the shared task might boost research is by creating public benchmark data, translations by many state-of-the-art systems and judgements of translation quality by humans (see also §9.4 on ways we are adding value).

3 Tracks and submission procedure

We offered two translation directions ("tracks"): translation from Swiss German Sign Language (DSGS) to German and vice versa.

Translation from DSGS to German was our primary translation direction in the sense that submitted systems were ranked on a leaderboard and we provided baseline systems. Systems translating from German to DSGS were not ranked on the leaderboard while the task was running, but we still encouraged participants to submit such systems. We were prepared to provide human evaluation for all submitted systems, regardless of the translation direction.

We deliberately did not limit the shared task to any particular kind of SL representation as input or output of an MT system. For DSGS-to-German translation participants were free to use video frames, pose estimation or something else. For German-to-DSGS participants were free to submit a video showing pose estimation output, an avatar or a photo-realistic signer.

Participants submitted translations on the OCELoT platform[5] which has a public leaderboard. We modified OCELoT slightly in order to disable automatic metrics on the leaderboard for German-to-DSGS, since currently no automatic metrics exist for SL output. Participants were allowed to make up to seven submissions, one of them the primary submission.

Main outcome Seven teams (including one from the University of Zurich whose submission we consider a baseline) participated in our task. All of them submitted to the DSGS-to-German track, while there were no submissions for the second translation direction.

[5] `https://ocelot-wmt22.azurewebsites.net/`

		SRF		FocusNews		Total
	direction	**episodes**	**segments**	**episodes**	**segments**	**segments**
training	(both)	29	7071	197	10136	17207
development	(both)	1	287	3	133	420
test	DSGS-to-German	1	242	5	246	488
	German-to-DSGS	1	183	5	228	411

Table 1: Overview of training, development and test data. SRF and FocusNews are two different training corpora (§4.2). Segment count for the training corpora is after automatic sentence segmentation. The development data for both translation directions is identical, while the test data is different for DSGS-DE and DE-DSGS.

4 Data

For this task we provided separate training, development and test data, where the training data was available from the beginning while the development and test data were released in several stages.

Table 1 gives a high-level overview of our training, development and test data.

Necessity of creating training data The data we provided are new corpora that we built and published. This was necessary because existing datasets for SL machine translation did not meet our requirements. Existing datasets either have a license that is too restrictive, are not parallel enough in the sense of being only "comparable corpora", are too small or have a very limited linguistic domain. For example, the most widely used dataset in SL machine translation research, PHOENIX (introduced in §2.4), has a size of 8k sentence pairs and contains only weather reports.

Following the long history of WMT shared tasks for spoken language machine translation (Akhbardeh et al., 2021), we opted for data that contains general news, hence a more open domain.

4.1 Licensing and attribution

Our training corpora have different licenses that are summarized here. This overview paper must be cited if the corpora are used.

FocusNews corpus This dataset can be used only for this shared task or its future iterations. Other uses of the data require express permission by the data owners. Interested parties should contact the organizers for further information.

SRF corpus This dataset can be used for non-commercial research under an Attribution-NonCommercial-ShareAlike 4.0 International license (CC BY-NC-SA 4.0)[6].

4.2 Training Data

The training data comprises two corpora, called FocusNews and SRF. The linguistic domain of both corpora is general news, and both contain parallel data between DSGS and German. The corpora are distributed through Zenodo[7].

The statistics of the two corpora are summarised in Table 2.

Training corpus 1: FocusNews [8] The FocusNews data originates from a former deaf online TV channel, FocusFive[9]. We provide the news episodes (FocusNews), as opposed to other programs. The data consists of 197 videos with associated subtitles of approximately 5 minutes each. The videos feature deaf signers of DSGS and represent the source for translation. The German subtitles were created in post-production by hearing SL interpreters.

We provide episodes within the time range of 2008 (starting with episode 43) to 2014 (up to episode 278). The videos were recorded with different framerates, either 25, 30 or 50 fps. The video resolution is 1280 x 720.

While this data set is small (by today's standards in spoken language machine translation), we emphasize the importance of using deaf signer data for shared tasks like ours. There are crucial differences between the signing of hearing interpreters and deaf signers, and interpreted signing

[6] https://creativecommons.org/licenses/by-nc-sa/4.0/

[7] https://zenodo.org/

[8] Here we describe Zenodo release version 1.3 of the corpus.

[9] https://www.youtube.com/c/focusfivetv

	FocusNews (release 1.3)	**SRF (release 1.2)**
Number of episodes	197	29
Time span of episodes	2008 – 2014	March 2020 – March 2021
Length of 1 episode	∼ 5 minutes	∼ 30 minutes
Number of signers	12	3
Signer status	deaf	hearing
Signing mode	Live signing from teleprompter (showing German text or glosses)	Live sign language interpretation
Translation source	DSGS	German
Total duration videos	19 hours	16 hours
Video resolution	1280 × 720	1280 × 720
Video framerate	25, 30 or 50	25
Number of parallel subtitles*	9943 / 10136	14265 / 7071
Number of monolingual subtitles*	(none)	883754 / 577418
Subtitle format	SRT	SRT
Sentence segmentation	automatic	manual
Subtitling mode	In post-production, after signing is already recorded	Pre-produced or live subtitles (using respeaking with ASR)

Table 2: Data statistics and characteristics of our training corpora. *= before / after automatic sentence segmentation.

may bear more resemblance to spoken language structures (Janzen, 2005).

Training corpus 2: SRF [10] The dataset contains daily national news and weather forecast episodes broadcast by the Swiss National TV (*Schweizerisches Radio und Fernsehen*, SRF)[11]. The episodes are narrated in Standard German of Switzerland (different from Standard German of Germany, and different from Swiss German dialects) and interpreted into DSGS. The interpreters are hearing individuals, some of them children of deaf adults (CODAs).

The subtitles are partly preproduced, partly created live via respeaking based on automatic speech recognition.

While both the subtitles and the signing are based on the original speech (audio), due to the live subtitling and live interpreting scenario, a temporal offset between audio and subtitles as well as audio and signing is inevitable. This offset or "alignment shift" is visualized in Figure 1.

Manual alignment In our training corpus, the offset between the signing and the subtitles was manually corrected by deaf signers with a good command of German. The live interview and weather forecast parts of each episode were ignored, as the quality of the subtitles tends to be noticeably lower for these parts.

The parallel data comprises 29 episodes of approximately 30 minutes each with the SL videos (without audio track) and the corresponding subtitles. We selected episodes from two time spans: 13/03/2020 to 19/06/2020 and 04/01/2021 to 26/02/2021, featuring three different SL interpreters. (Three interpreters consented to having their likeness used for this shared task.) The videos have a framerate of 25 fps and a resolution of 1280 x 720.

In addition to the parallel data we provided all available German subtitles from 2014 to 2021 as monolingual data. In total, there are 1949 subtitle files with a total of 570k sentences (count after automatic segmentation).

4.3 Development data

The development data consists of segments extracted from undisclosed SRF and FocusNews episodes (see §4.2 for a general description). This data was also manually aligned and the signer is a "known" person that appeared in the training set. The framerate of development videos is 25 fps for SRF and 50 fps for FocusNews.

4.4 Test data

We distribute separate test data for our two translation directions.

[10] Here we describe Zenodo release version 1.2 of the corpus. The data provided here is an extended version of the dataset published as part of the Content4All project (EU Horizon 2020, grant agreement no. 762021).

[11] https://www.srf.ch/play/tv/sendung/tagesschau-in-gebaerdensprache?id=c40bed81-b150-0001-2b5a-1e90e100c1c0

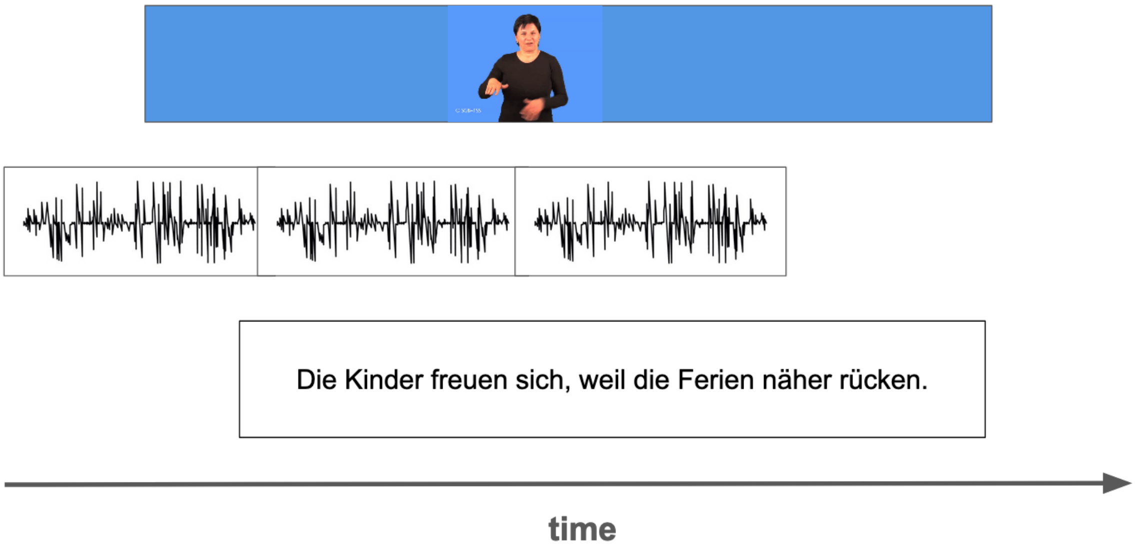


Figure 1: Illustration of alignment shift in sign language corpora. From top to bottom: a sign language video, an audio track with speech, a spoken language subtitle in German. Information in these three modalities do not start and end at the same time, adjusting their start and end times is referred to as *alignment*.

DSGS-to-German Additional, undisclosed SRF and FocusNews episodes that are manually aligned. As for the development data, the signers are "known" persons and the framerate of videos is 25 fps for SRF and 50 fps otherwise.

German-to-DSGS This subset of the test data has two distinct parts:

1. Additional, undisclosed FocusNews episodes that are manually aligned. As for the development data, the signers are "known" persons and the framerate of videos is 50 fps.

2. New translations created specifically for this shared task. The domain is identical to the training data (general news). In this case German subtitles are the source for human translation, DSGS videos are the target. The human translator is deaf (in contrast to all of the SRF data, where signers are hearing interpreters). The framerate of these videos is 50 fps and they are recorded with a green screen.

For German-to-DSGS translation we consider it important that the reference translations are created by deaf signers instead of hearing interpreters.

4.5 Automated access to training data

Our baseline system described in §6.1 automatically downloads all subsets of the data.

In addition, we added our training corpora to the Sign Language Datasets library (Moryossef and Müller, 2021b). The datasets can now be loaded automatically as a Tensorflow data set, provided that the user has previously obtained Zenodo access tokens.

5 Data preprocessing

For each data set described in §4 we provided videos and corresponding subtitles. In addition, we included pose estimates (location of body keypoints in each frame) as a convenience.

5.1 Video processing

Videos are re-encoded with lossless H264 and use an mp4 container. The framerate of videos is unchanged, meaning either 25, 30 or 50. We are not distributing the original videos but ones that are preprocessed in a particular way so that they only show the part of each frame where the signer is located (cropping) and the background is replaced with a monochrome color (signer masking), see Figure 2 for examples.

Cropping We manually annotate a rectangle (bounding box) around where the signer is located for each video. We then crop the video to only keep this region using the FFMPEG library.

Signer segmentation and masking To the cropped video we apply an instance segmentation

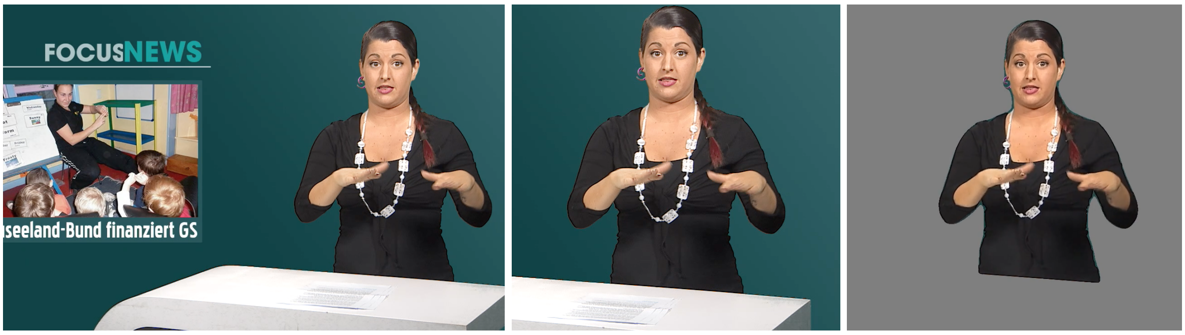


Figure 2: Illustration of video preprocessing steps (cropping, instance segmentation and masking). From left to right: original frame, cropped frame, masked frame.

model, Solo V2 (Wang et al., 2020), to separate the background from the signer. This produces a mask that can be superimposed on the cropped video to replace each background pixel in a frame with a grey color (`[127,127,127]` in RGB).

5.2 Subtitle processing

For subtitles that are not manually aligned (all of FocusNews and monolingual SRF data), automatic sentence segmentation is used to redistribute text across subtitle segments, see Figure 3 for examples.

This process also adjusts timecodes in a heuristic manner if needed. For instance, if automatic sentence segmentation detects that a well-formed sentence stops in the middle of a subtitle, a new end time will be computed. The end time is proportional to the location of the last character of the sentence, relative to the entire length of the subtitle. See Example 2 in Table 3 for an illustration of this case.

5.3 Pose processing

"Poses" are an estimate of the location of body keypoints in video frames. The exact set of keypoints depends on the pose estimation system, well known ones are OpenPose (Cao et al., 2019)[12] and MediaPipe Holistic (Lugaresi et al., 2019)[13]. Usually such a system provides 2D or 3D coordinates of keypoints in each frame, plus a confidence value for each keypoint.

The input for pose processing are cropped and masked videos (§5.1). See Figure 3 for examples of pose estimation on our data.

OpenPose We are using the OpenPose Body135 model. OpenPose often detects several people in our videos, even though there is only one single person present. We distribute the original predictions which contain all people that OpenPose detected.

MediaPipe Holistic As an alternative, we also estimate signers' poses with the MediaPipe Holistic system developed by Google. Unlike our OpenPose model, which only provides 2D joint locations, MediaPipe produces both 2D and 3D joint location coordinates. Values from Holistic are normalized between 0 and 1, instead of referring to actual video coordinates.

6 Baseline and submitted systems

In this section we describe all submissions to our shared task. In case there are substantial differences between the primary and secondary submissions of a team we opted to describe the primary submission here. At the time of writing this overview paper six out of seven teams have given us detailed information about their submissions. The submissions are summarized in Table 4.

Overall, the participating teams have diverse academic backgrounds, most of them combine computer vision and NLP expertise. All submitted systems are sequence-to-sequence models based on Transformers (Vaswani et al., 2017). Participants chose to represent sign language data as either video frames (using a visual feature extractor on the encoder side) or pose features, with no clear majority in this regard.

Two systems, by LATTIC and MSMUNICH, are unconstrained because their visual encoder component is pretrained on WSASL (Li et al., 2020) or is an existing model taken from Varol

[12] `https://github.com/CMU-Perceptual-Computing-Lab/openpose`

[13] `https://ai.googleblog.com/2020/12/mediapipe-holistic-simultaneous-face.html`

Example 1	
Original subtitle	After automatic segmentation
81 00:05:22,607 -> 00:05:24,687 Die Jury war beeindruckt 82 00:05:24,687 -> 00:05:28,127 und begeistert von dieser gehörlosen Frau.	48 00:05:22,607 -> 00:05:28,127 Die Jury war beeindruckt und begeistert von dieser gehörlosen Frau.
Example 2	
Original subtitle	After automatic segmentation
7 00:00:24,708 -> 00:00:27,268 Die Invalidenversicherung Region Bern startete 8 00:00:27,268 -> 00:00:29,860 dieses Pilotprojekt und will herausfinden, ob man es 9 00:00:29,860 -> 00:00:33,460 zukünftig umsetzen kann. Es geht um die Umsetzung	4 00:00:24,708 -> 00:00:31,720 Die Invalidenversicherung Region Bern startete dieses Pilotprojekt und will herausfinden, ob man es zukünftig umsetzen kann.

Table 3: Examples of automatic sentence segmentation for German subtitles. The subtitles are formatted as SRT, a common subtitle format.

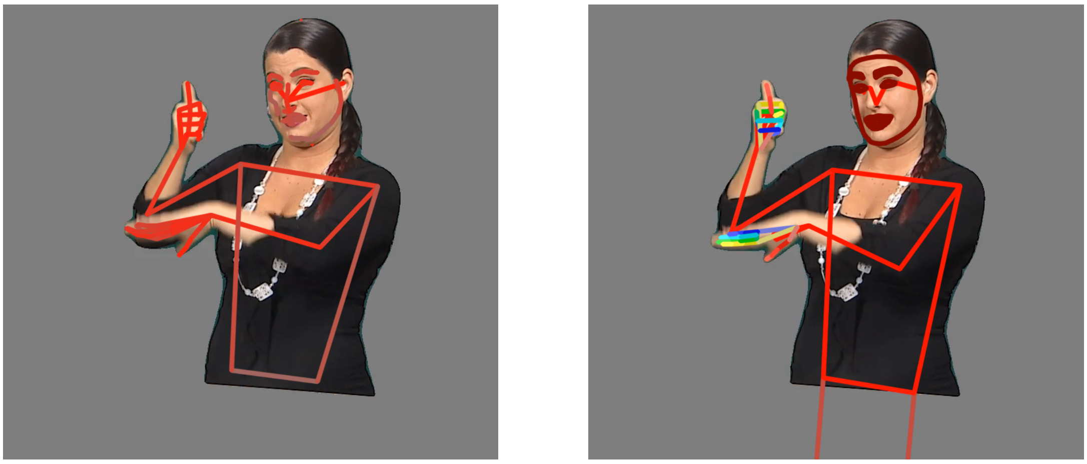


Figure 3: Examples of the output of pose estimation systems overlaid over the original video frames. Left: OpenPose, right: MediaPipe Holistic.

	BASELINE	LATTIC	MSMUNICH	UPC	DFKI-SLT	DFKI-MLT	NJUP-MTT
Constrained	✔	-	-	✔	✔	✔	?
Multilingual	-	-	-	-	-	-	?
Document-level	-	-	-	-	-	-	?
Model ensemble	-	-	-	-	-	-	?
Pretrained components	-	✔	✔	-	-	-	?
Monolingual data	-	-	-	-	-	-	?
Synthetic data	-	-	-	-	✔	-	?
Signed language representation	OP	Video frames	Video frames	MH	MH	Video frames	?
Spoken language representation	SP	SP	other^1	SP	other^2	-	?
Open-source code	✔	(✔)	-	✔	✔	(✔)	?

Table 4: Overview of characteristics of submitted systems. NJUP-MTT did not disclose any information. In the code row, checkmarks are clickable links. OP=OpenPose, MH=MediaPipe Holistic, SP=Sentencepiece, (✔)=authors plan to publish the code, other^1=text is normalized, but not segmented, other^2=text is lowercased, but not segmented

et al. (2021). Only one team (DFKI-SLT) used synthetic parallel data and no submission used the monolingual subtitles we distributed.

Three teams have published their code, with two other teams planning to do so in the future.

6.1 Submission by UZH (baseline system)

We provided code to train baseline systems for DSGS to German in a public Github repository (Müller et al., 2022)[14]. The codebase contains scripts to preprocess data, train, translate and evaluate models and should allow to reproduce our results exactly.

The underlying sequence-to-sequence toolkit is Sockeye (Hieber et al., 2022) which is based on Pytorch (Paszke et al., 2019). We adapted Sockeye so that it supports encoding or decoding continuous vectors instead of discrete sequences of tokens. Our system is a pose-to-text translation model that reads a sequence of pose frames and converts them to the model size with a simple learned projection. The baseline does not involve pretraining or additional data and is therefore a constrained submission.

Preprocessing We used OpenPose (Cao et al., 2019) predictions (as opposed to MediaPipe Holistic or a third option). If OpenPose predicted several people in a frame, we simply chose the first one and ignored all other values. Poses are normalized by shoulder width. We convert all pose sequences to a framerate of 25 fps. On the spoken language side we do not apply any preprocessing except learning and applying a Sentencepiece segmentation model (Kudo, 2018) with a vocabulary size of 1000.

For training and translation we used one Tesla V100-32GB GPU and the training took between two and four hours.

6.2 Submission by LATTIC (Shi et al., 2022)

The system submitted by LATTIC is a Transformer-based sequence-to-sequence model which uses as input visual representations derived from an Inflated 3D ConvNet (I3D) (Carreira and Zisserman, 2017) and text as the target. The I3D models is pretrained on the WLASL[15] dataset (an isolated sign dataset). The input representation is resized video frames, the frames were resized to 224x224. For the spoken language side Sentencepiece (Kudo and Richardson, 2018) was used to generate a vocabulary of 18k tokens. The system is developed from scratch, without the use of existing MT software, and has a Transformer architecture (Vaswani et al., 2017). The I3D model is first trained on Kinetics, an action recognition dataset (Carreira and Zisserman, 2017), then it is trained for isolated sign language recognition. Before feeding input to the model, each isolated sign video is truncated, resized, randomly cropped to 224x224 and horizontally flipped with probability 0.5. Models were trained on several GPU types (A4000, A6000 and Titan RTX) and the training took roughly four hours per model.

6.3 Submission by MSMUNICH (Dey et al., 2022)

Microsoft's submission to WMT-SLT is a sequence-to-sequence Transformer model. It is based on an existing model pretrained on the

[14] https://github.com/bricksdont/sign-sockeye-baselines

[15] https://github.com/dxli94/WLASL

BSL1K dataset (Varol et al., 2021)[16]. Similar to the submission of LATTIC, this system also uses a pretrained I3D model. The system takes as input consecutive video frames and predicts over 1000 signs. For the text side, text normalisation such as lowercasing, conversion of numerals and data cleaning were applied. The authors emphasize that such careful data preprocessing and postprocessing was crucial. The underlying MT framework is Fairseq (Ott et al., 2019).

6.4 Submission by SLT-UPC (Tarrés et al., 2022)

The submission of UPC[17] is also a Transformer-based sequence-to-sequence model, based on a smaller Transformer architecture. To pretrain the model, PHOENIX (Forster et al., 2014) data was used. However, the results achieved with pretraining were no better than the primary submission (without pretraining). The authors built independent vocabularies for each training corpus. The best results were obtained by only training on the FocusNews dataset.

As a representation for the SL side, MediaPipe Holistic was used, re-extracting the features using the pose library by Moryossef and Müller (2021a). The authors interpolated the pose sequences to unify the framerate to 25fps and used data augmentation on the poses (using pose libary augmentation functions such as rotation, scaling and shear). For the text side Sentencepiece was used to generate vocabularies of 1000, 2000 and 4000. Their main submission had a vocabulary of 1000. The code is based on Fairseq and is available on GitHub[18]. To train their models, one Nvidia GeForce RTX 3090 was used and training for the main submission took roughly 3.5 hours.

6.5 Submission by DFKI-SLT (Hufe and Avramidis, 2022)

The submission of DFKI-SLT is a sequence-to-sequence model trained with JoeyNMT (Kreutzer et al., 2019), using chrF as the validation metric. The authors describe their system as having three main modules. In the first, SL images are converted into intermediate pose keypoint representations; the second module employs data augmentation (geometrical transformations) to increase sample efficiency and decrease the effect of spurious feature correlations; and the third employs a Transformer network to perform translation.

The system is trained only on FocusNews. The representation of the SL side was based on MediaPipe Holistic. The text side was only lowercased and the maximum sentence length was set to 400. The models were trained on an Nvidia RTXA6000.

6.6 Submission by DFKI-MLT (Hamidullah et al., 2022)

The main idea behind the DFKI-MLT approach is to learn feature representation and translation in a single model, and to train them together. The system architecture consists of two connected blocks: the first block, implemented using CNNs, is intended to capture visual representations and the second one, implemented with Transformers, aims to capture language. The visual component is based on a ResNet (Hara et al., 2017). In particular, the visual encoding in the submitted system consists of the original 3D ResNet10 with output conversion. The conversion creates a sequence of vectors from the single output vector to adapt to the Transformer encoder input. The visual vector is projected through a linear layer which is connected directly to the language block. The language block is a simple Transformer. The training is end-to-end, aiming to force the visual block to take into account the language representation when building the visual embedding.

6.7 Submission by NJUPT-MTT

Finally, we received submissions from the machine translation lab at Nanjing University of Posts and Telecommunications (NJUPT-MTT). No system paper was submitted and the authors did not provide further information.

7 Evaluation Protocols

We performed both a human (§7.1) and an automatic (§7.2) evaluation of translation quality. Our final system ranking is based on the human evaluation only.

7.1 Human evaluation

In our human evaluation, we followed the setting established by the recent WMT21 conference (Akhbardeh et al., 2021) and adapted it to the requirements of SLT evaluation.

[16] https://www.robots.ox.ac.uk/~vgg/research/bslattend/

[17] https://www.upc.edu/ca

[18] https://github.com/mt-upc/fairseq/tree/wmt-slt22

We employed the source-based direct assessment (DA; Graham et al., 2013; Cettolo et al., 2017) methodology with document context, extended with Scalar Quality Metric (SQM; Freitag et al., 2021), which was piloted at the IWSLT 2022 evaluation campaign (Anastasopoulos et al., 2022). Assessments were performed on a continuous scale between 0 and 100 as in traditional DA but with 0-6 markings on the analogue slider and custom annotator guidelines specifically designed for our task.

Human evaluation settings We used the Appraise evaluation framework[19] (Federmann, 2018) for collecting segment-level judgements within document context. As there were submissions in the DSGS-to-German direction only (§3), we only set up a sign-to-text human evaluation campaign. Annotators were presented with video fragments as source context and translation outputs of a random document from an MT system. The reference translation and the official baseline were included as additional system outputs. Documents longer than ten segments were split into document snippets with ten or fewer consecutive segments. A screenshot of an example annotation in Appraise is presented in Figure 4.

We hired four evaluators who were native German speakers and trained DSGS interpreters. They did not have prior experience with evaluation of MT output. Each evaluator was assigned an identical set of annotation tasks comprising documents from the entire test set and all participating systems, including the baseline system and the reference translation. 196 segments were given to each annotator more than once to conform to Appraise's requirement of 100 segments per task and in order to measure intra-annotator agreement.

We did not include any quality control items in the annotation tasks as we had multiple independent annotations of the entire test set and because of the very low quality of translations, which would make them indistinguishable from segments with randomly replaced words or phrases used as quality control items.

Justification for custom guidelines We designed custom guidelines to account for different modalities (e.g. avoid confusing mentions of "text" in the instructions when the source or target is in fact a video) and to tailor them towards SL content. For example, we added *naturalness of motion* as an evaluation criterion for evaluations with SL output. Following IWSLT 2022, we also removed any mention of "grammar" to shift emphasis away from grammatical issues towards translation-breaking differences in meaning. The full instructions to evaluators in English and German are listed in Appendix A.

Data and scripts used for generating tasks and computing the final system rankings are publicly available in a Github repository.[20]

7.2 Automatic evaluation

To complement our human evaluation (which provides the main ranking) we also provide an automatic evaluation. We evaluate the submissions and the baseline system from DSGS into German using three automatic metrics: BLEU (Papineni et al., 2002), chrF (Popović, 2015) and BLEURT (Sellam et al., 2020). We note that learned, semantic metrics correlate better with human judgement (Kocmi et al., 2021), but if they consider the source text as an input (e.g. COMET; Rei et al., 2020), they cannot be used in our context because our source is video and not text. We use sacreBLEU (Post, 2018) for BLEU[21] and chrF[22] and the python library for BLEURT.[23] In all cases, we estimate 95% confidence intervals via bootstrap resampling (Koehn, 2004) with 1000 samples.

8 Results

8.1 Human evaluation

Assessment scores Three out of the four evaluators completed all tasks, which gave us at least three independent judgements for each segment from the official test set. In total, for the output of eight systems, we collected 133,000 segment-level and 1,191 document-level assessment scores, which averages to 1,811.4 scores per system.

System ranking The system ranking is based on the average DA segment-level scores computed from the human assessment scores. We did not

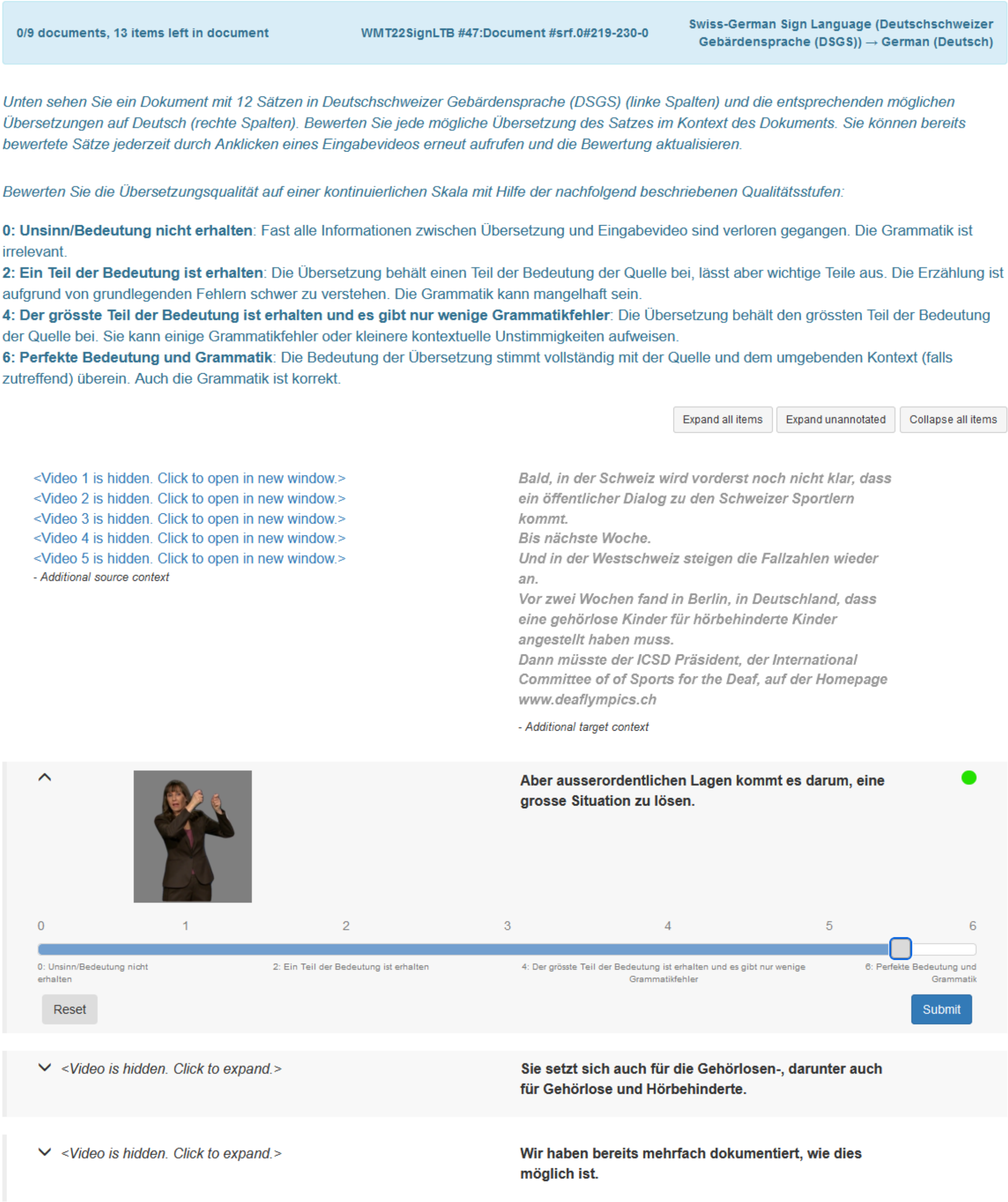
[19] `https://github.com/AppraiseDev/Appraise`

[20] `https://github.com/WMT-SLT/wmt-slt22`

[21] `BLEU|nrefs:1|bs:1000|seed:12345|case:mixed|eff:no|tok:13a|smooth:exp|version:2.2.0`

[22] `chrF2|nrefs:1|bs:1000|seed:12345|case:mixed|eff:yes|nc:6|nw:0|space:no|version:2.2.0`

[23] BLEURT v0.0.2 using checkpoint BLEURT-20.



0/9 documents, 13 items left in document

WMT22SignLTB #47:Document #srf.0#219-230-0

Swiss-German Sign Language (Deutschschweizer Gebärdensprache (DSGS)) → German (Deutsch)

Unten sehen Sie ein Dokument mit 12 Sätzen in Deutschschweizer Gebärdensprache (DSGS) (linke Spalten) und die entsprechenden möglichen Übersetzungen auf Deutsch (rechte Spalten). Bewerten Sie jede mögliche Übersetzung des Satzes im Kontext des Dokuments. Sie können bereits bewertete Sätze jederzeit durch Anklicken eines Eingabevideos erneut aufrufen und die Bewertung aktualisieren.

Bewerten Sie die Übersetzungsqualität auf einer kontinuierlichen Skala mit Hilfe der nachfolgend beschriebenen Qualitätsstufen:

0: Unsinn/Bedeutung nicht erhalten: Fast alle Informationen zwischen Übersetzung und Eingabevideo sind verloren gegangen. Die Grammatik ist irrelevant.
2: Ein Teil der Bedeutung ist erhalten: Die Übersetzung behält einen Teil der Bedeutung der Quelle bei, lässt aber wichtige Teile aus. Die Erzählung ist aufgrund von grundlegenden Fehlern schwer zu verstehen. Die Grammatik kann mangelhaft sein.
4: Der grösste Teil der Bedeutung ist erhalten und es gibt nur wenige Grammatikfehler: Die Übersetzung behält den grössten Teil der Bedeutung der Quelle bei. Sie kann einige Grammatikfehler oder kleinere kontextuelle Unstimmigkeiten aufweisen.
6: Perfekte Bedeutung und Grammatik: Die Bedeutung der Übersetzung stimmt vollständig mit der Quelle und dem umgebenden Kontext (falls zutreffend) überein. Auch die Grammatik ist korrekt.

<Video 1 is hidden. Click to open in new window.>
<Video 2 is hidden. Click to open in new window.>
<Video 3 is hidden. Click to open in new window.>
<Video 4 is hidden. Click to open in new window.>
<Video 5 is hidden. Click to open in new window.>
- Additional source context

Bald, in der Schweiz wird vorderst noch nicht klar, dass ein öffentlicher Dialog zu den Schweizer Sportlern kommt.
Bis nächste Woche.
Und in der Westschweiz steigen die Fallzahlen wieder an.
Vor zwei Wochen fand in Berlin, in Deutschland, dass eine gehörlose Kinder für hörbehinderte Kinder angestellt haben muss.
Dann müsste der ICSD Präsident, der International Committee of of Sports for the Deaf, auf der Homepage www.deaflympics.ch
- Additional target context

Aber ausserordentlichen Lagen kommt es darum, eine grosse Situation zu lösen.

0
1
2
3
4
5
6
0: Unsinn/Bedeutung nicht erhalten
2: Ein Teil der Bedeutung ist erhalten
4: Der grösste Teil der Bedeutung ist erhalten und es gibt nur wenige Grammatikfehler
6: Perfekte Bedeutung und Grammatik
Reset
Submit

<Video is hidden. Click to expand.>

Sie setzt sich auch für die Gehörlosen-, darunter auch für Gehörlose und Hörbehinderte.

<Video is hidden. Click to expand.>

Wir haben bereits mehrfach dokumentiert, wie dies möglich ist.

Figure 4: A screenshot of an example sign-to-text annotation task in Appraise featuring document-level source-based direct assessment (DA) with scalar quality metrics (SQM) and custom annotator guidelines in German.

make any distinction between segment-level and document-level scores, simply including the latter as additional data for computing the average scores.

The official system ranking is presented in Table 5. Systems which significantly outperform all others, according to Wilcoxon rank-sum test $p < 0.05$, are grouped into clusters, which is indicated by horizontal lines. Rank ranges giving an indication of the respective system's translation quality within a cluster are based on the same head-to-head statistical significance tests. Contrary to previous evaluation campaigns (Akhbardeh et al., 2021) which calculate the rankings based on standardized scores (z-scores), we decided to not do so, because the large number of zero-scored items led to a rather skewed standardization scale which affected the calculation of the clusters.

According to our human evaluation (Table 5), MSMUNICH and LATTIC have the highest quality score among all MT systems. All other systems ended up in the same cluster with overall lower translation quality. Both winning systems are unconstrained, having been pretrained on other SL datasets, and achieve an average score of 2 in the continuous range of [0, 100], as compared to a a score of 87 for human translations and 0.52 for the baseline system. By looking at the domain-specific results, however, one can see that the performance of these two systems is around 3.5 for the FocusNews part of the test set and only 0.28-0.38 for the SRF part.

We show an additional analysis of the score distribution for each system in Appendix D.

Annotator agreement In Table 6 we are reporting intra-annotator agreement, measured with Fleiss κ (Fleiss, 1971) only as an approximation, noting the concerns of Ma et al. (2017) that kappa coefficients are not suitable for continuous scales. In order to calculate the coefficient, the values have been discretized in seven bins in the scale 0-6, since those were the scores marked on the continuous evaluation bar that was given to the annotators. One can observe that the intra-annotator agreement for raters 1 and 2 is *good* whereas for raters 3 and 4 is *very good* (Landis and Koch, 1977; Agresti, 1996).

In order to ensure the agreement between the annotators, we computed the ranks with different combinations of annotators and we did not observe changes in the ranks.

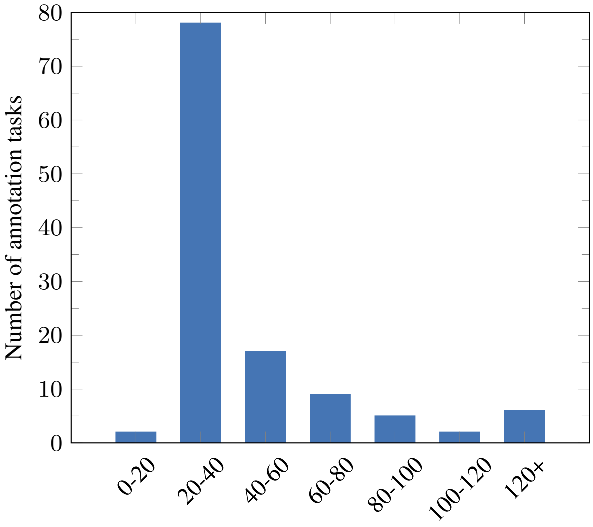


Figure 5: Number of task completion times (a task consists of 100 segments) grouped into 20-minute buckets, after removing top and bottom 5-percentiles.

Evaluation speed Three evaluators have completed the entire evaluation. A single task requiring 100 segment-level and about 12 document-level annotations took on average 45 minutes to complete, after excluding 5% of slowest and fastest task annotations. The majority of tasks were finished in between 20 and 40 minutes as shown in Figure 5.

On average, evaluators judged with a speed of 200 to 250 sentence pairs per hour. This is in line with previous evaluations for spoken language MT. We believe having such an estimate of evaluation speed is useful for future evaluations.

Feedback from evaluators After completing the evaluation two out of four evaluators filled in a form meant for feedback regarding the evaluation procedure and the Appraise platform. All evaluators gave us additional informal feedback.

In general, evaluators reported that their experience with Appraise was positive, and that our instructions were clear. At least two people would be willing to do similar work in the future. Concerning Appraise development, at least two people experienced technical problems[24] and evaluators suggested that the user interface could be improved in some places. For instance, automatically playing videos could make evaluations more efficient.

[24] During the evaluation period there were major outages on Azure and the technical issues reported by our evaluators may be unrelated to the user interface or evaluation task.

all			SRF			FN		
Rank	Ave.	System	Rank	Ave.	System	Rank	Ave.	System
1	87.051	HUMAN	1	87.051	HUMAN	1	93.568	HUMAN
2-3	2.075	MSMUNICH	2-3	2.075	MSMUNICH	2-3	3.833	MSMUNICH
2-3	2.008	SLATTIC	2-3	2.008	SLATTIC	2-3	3.610	SLATTIC
4-5	0.520	UZH (baseline)	4-5	0.520	UZH (baseline)	4-6	1.028	UZH (baseline)
4-8	0.437	DFKI-MLT	4-8	0.437	DFKI-MLT	4-7	0.853	DFKI-MLT
5-8	0.339	DFKI-SLT	5-8	0.339	DFKI-SLT	4-7	0.671	DFKI-SLT
5-8	0.207	UPC	5-8	0.207	UPC	5-8	0.407	UPC
5-8	0.041	NJUPT-MTT	5-8	0.041	NJUPT-MTT	7-8	0.033	NJUPT-MTT

Table 5: Official results of the WMT22 Sign Language Translation task for translation from Swiss German Sign Language to German. Systems are ordered by averaged (non-standardized) human score in the percentage scale. Lines indicate clusters according to a Wilcoxon rank-sum test $p < 0.05$. Gray rows indicate unconstrained systems.

annotator	κ	**items**
1	0.77±0.07	235
2	0.76±0.13	62
3	0.90±0.06	235
4	0.88±0.06	235

Table 6: Intra-annotator agreement based on the Fleiss κ coefficient for reliability of agreement (with scores discretized in the scale 0-6).

Informally, evaluators have told us that some videos do not have ideal cuts, in the sense that the beginning or end are slightly cut off. This is perhaps inevitable in continuous signing, or a problem in our manual alignment process. They have also pointed out that showing machine-translated target context can be confusing because for our use case quality is so low.

More detailed feedback forms submitted by evaluators are listed in Appendix C.

8.2 Automatic Evaluation

Table 7 summarises the results of the automatic evaluation. We report the scores for the full test set and also for the SRF and FocusNews subsets and boldface the primary submissions that have been evaluated manually. The low scores for all systems and metrics demonstrate the difficulty of the task. For most systems but SLATTIC with BLEU, translation quality is higher for FocusNews than for SRF. This might be an effect of the length of the source videos: SRF videos are six times longer than FocusNews, which might make the alignment with the textual part more difficult at sentence level.

The best system in the automatic evaluation depends on the evaluation metric. MSMUNICH.2 is the best system according to BLEU, SLATTIC.4 according to chrF and MSMUNICH.1 according to BLEURT. Notice that only the best system according to BLEU among these three was submitted as primary system and therefore manually evaluated. This shows that participants probably used mainly BLEU as the metric for development, except DFKI-SLT who reported that they used chrF because BLEU was always zero.

The correlation between human rankings and automatic metrics is delicate because we only have seven data points. The metric that correlates best with human scores at system level is BLEU ($r = 0.510$, $\rho = 0.571$) followed by chrF ($r = 0.508$, $\rho = 0.214$). BLEURT shows only a weak correlation with $r = 0.314$ and $\rho = 0.286$. In our scenario, translation quality is really low, and the sentences that have been properly translated are very short (e.g. *Bis nächste Woche.*). In this case, n-gram matching metrics perform better than semantic metrics.

See Appendix B for an extended discussion of the correlation between all automatic metrics (BLEU, chrF, BLEURT).

9 Discussion

9.1 General translation quality

Overall, all systems perform poorly in our shared task, as there is an extreme difference in average score between all systems and the human reference translation. The systems exhibit well-known problems of natural language generation such as overfitting to few high-probability hypotheses and hallucination (Lee et al., 2018; Raunak et al., 2021).

The best submitted system in the best case achieves an average score of about 4 out of 100, which indicates that current automatic translations are not usable in practice, unlike spoken language MT where in specific scenarios experiments have

	BLEU			**chrF**			**BLEURT**		
Submission	all	SRF	FN	all	SRF	FN	all	SRF	FN
UZH (baseline)	0.12±0.06	0.09±0.03	0.19±0.11	5.5±0.5	5.2±0.5	5.8±0.8	0.102±0.006	0.095±0.006	0.110±0.009
DFKI-SLT	0.08±0.01	0.10±0.04	0.11±0.02	18.2±0.4	17.9±0.5	18.7±0.6	0.109±0.006	0.093±0.004	0.122±0.009
DFKI-MLT.1	0.07±0.05	0.05±0.02	0.12±0.10	6.6±0.5	6.4±0.6	6.9±0.6	0.100±0.008	0.097±0.009	0.100±0.012
DFKI-MLT.2	0.11±0.06	0.08±0.03	0.17±0.13	6.8±0.5	7.0±0.7	6.5±0.7	0.083±0.008	0.074±0.008	0.091±0.013
DFKI-MLT.3	0.08±0.04	0.06±0.02	0.13±0.10	6.5±0.5	6.8±0.8	6.2±0.7	0.075±0.009	0.067±0.009	0.081±0.014
DFKI-MLT.4	0.02±0.01	0.02±0.01	0.04±0.02	3.6±0.2	3.4±0.3	3.8±0.3	0.066±0.004	0.063±0.004	0.070±0.008
DFKI-MLT.5	0.04±0.02	0.03±0.00	0.08±0.04	5.4±0.3	5.1±0.3	5.6±0.4	0.078±0.004	0.074±0.005	0.080±0.007
MSMUNICH.1	0.44±0.21	0.34±0.18	0.63±0.35	17.1±0.5	16.3±0.7	17.8±0.9	0.166±0.013	0.147±0.012	0.179±0.022
MSMUNICH.2	0.56±0.30	0.28±0.13	0.84±0.51	17.4±0.5	17.0±0.5	17.9±0.8	0.150±0.011	0.132±0.008	0.163±0.019
NJUPT-MTT.1	0.09±0.01	0.13±0.03	0.13±0.03	14.6±0.5	14.8±0.7	14.4±0.8	0.127±0.006	0.125±0.007	0.130±0.009
NJUPT-MTT.2	0.10±0.01	0.13±0.03	0.14±0.03	14.1±0.5	14.2±0.7	14.0±0.7	0.117±0.006	0.117±0.007	0.117±0.009
SLATTIC.1	0.25±0.12	0.30±0.18	0.24±0.10	19.5±0.4	19.2±0.5	19.8±0.7	0.074±0.010	0.055±0.007	0.090±0.016
SLATTIC.2	0.20±0.14	0.32±0.23	0.10±0.02	17.9±0.5	17.4±0.7	18.5±0.8	0.092±0.012	0.080±0.010	0.098±0.017
SLATTIC.3	0.14±0.09	0.21±0.16	0.09±0.06	17.4±0.5	17.0±0.6	17.8±0.7	0.096±0.012	0.081±0.010	0.106±0.019
SLATTIC.4	0.19±0.15	0.28±0.23	0.11±0.02	19.9±0.5	19.9±0.6	19.8±0.8	0.088±0.011	0.067±0.006	0.107±0.019
SLATTIC.5	0.18±0.06	0.21±0.09	0.19±0.10	17.9±0.5	17.4±0.6	18.3±0.8	0.107±0.011	0.093±0.007	0.119±0.019
SLATTIC.6	0.07±0.03	0.15±0.07	0.04±0.01	15.0±0.4	14.8±0.5	15.0±0.6	0.103±0.010	0.094±0.006	0.110±0.017
SLT-UPC.1	0.34±0.22	0.29±0.14	0.43±0.33	15.6±0.6	15.4±0.8	15.8±0.8	0.131±0.005	0.126±0.006	0.136±0.008
SLT-UPC.2	0.35±0.21	0.29±0.14	0.43±0.30	16.2±0.6	15.4±0.8	17.0±0.9	0.136±0.004	0.126±0.006	0.145±0.007
SLT-UPC.3	0.41±0.33	0.24±0.10	0.54±0.47	15.5±0.6	15.1±0.8	16.0±0.9	0.144±0.006	0.131±0.006	0.157±0.010
SLT-UPC.4	0.28±0.09	0.26±0.11	0.37±0.16	12.2±0.4	12.3±0.6	12.1±0.6	0.113±0.004	0.122±0.006	0.103±0.006
SLT-UPC.5	0.24±0.10	0.32±0.14	0.25±0.12	12.0±0.4	12.1±0.6	11.9±0.5	0.102±0.004	0.110±0.006	0.094±0.006
SLT-UPC.6	0.28±0.09	0.26±0.11	0.37±0.16	12.2±0.4	12.3±0.6	12.1±0.6	0.113±0.004	0.122±0.006	0.103±0.006
SLT-UPC.7	0.50±0.26	0.37±0.13	0.61±0.38	12.3±0.5	11.9±0.7	12.7±0.8	0.111±0.006	0.110±0.007	0.111±0.011

Table 7: Automatic evaluation of all the submission for the full WMT-SLT test set (all), the SRF subset and the FocusNews (FN) subset. Mean and 95% confidence intervals obtained via bootstrap resampling are shown. Primary submissions manually evaluated are boldfaced. Note that the official ranking is given by the human evaluation (Table 5).

shown systems to be on par with human translation (Hassan et al., 2018; Popel et al., 2020). In the following paragraphs we discuss potential reasons for this outcome.

Size of training data The corpora we have built for this shared task (§4) are superior to existing datasets (in terms of size, license, linguistic domain and alignment quality), but are still small. Taken together our corpora contain 20k parallel sentence pairs only, and 600k monolingual German sentences. This limits the optimal translation quality that could in theory be obtained in a constrained setup. This is corroborated by the fact that the two unconstrained systems have won the shared task (§8).

Building larger parallel SL corpora in itself is challenging. Even though recently steps were taken to collect larger amounts of data (e.g. in the projects EASIER and SignON), such resources are not immediately useful because basic linguistic tools used to prepare parallel corpora are not available (§2.3, lack of basic linguistic tools). For spoken language NLP, such tools are commonplace, work well and are used to automatically compile large corpora. For example, Bitextor[25], a tool developed in the Paracrawl project (Bañón et al., 2020), relies on the automatic alignment tool BleuAlign (Sennrich and Volk, 2011).

Modality gap But even if much more training data was available, it is likely that current MT methods are not adapted well enough to SL data. NLP methods in general are tailored towards text and may perform worse or not be applicable at all to other modalities. For example, there are currently no efficient tools for automatic SL segmentation (Yin et al., 2021), while for text-based MT, subword segmentation (Sennrich et al., 2016; Kudo, 2018) has become a staple in research.

While all systems submitted this year are signed-to-spoken systems, the modality gap is more apparent for automatic spoken-to-sign translation because generating continuous outputs requires more fundamental changes to existing MT toolkits (as opposed to the changes necessary for continuous inputs).

[25] https://github.com/bitextor/bitextor

The proclivity of existing MT research for text data is confirmed by the number of recent works that chose to represent SL content as (textual) gloss sequences, despite the fact that glosses are not an adequate representation of meaning (Anonymous, 2022).

9.2 Reliability of evaluation procedure

Our evaluation is reliable since we conduct a human evaluation (compared to other shared tasks which produce official rankings based on automatic metrics). But even compared to shared tasks that do offer human evaluation (such as the General task this year), we believe that our evaluation is strong, since we have three to four (at least three) independent judgements for each system output across the entire test set.

9.3 Limitations of shared task setup

We note several limitations of the specific experimental setup in this year's shared task.

Generalization As explained in §4 all signers that appear in the development and test sets are known, in the sense of also being present in the training data. It is therefore important to emphasize that our shared task evaluates the performance of systems on familiar signers, and does not test generalization to unseen individuals.

Recording conditions Since our training data is derived from news broadcasts, the recording conditions and video quality are favourable. For example, the signer is always recorded against a monochrome and static background. The recording angle is very consistent, as cameras are mounted on a fixed rig. Signers always directly face the camera. The recording conditions therefore resemble laboratory conditions.

This means that our shared task does not evaluate "signing in the wild" (examples: mobile recordings of varying quality, varying angles, moving background including other people) and it is likely that the outcome would be different in that case.

Interpretation vs. translation Some of our training material is interpreted live (§4). Interpretation has constraints that are very different from offline translation, most notably, interpreters are under severe time pressure. This has consequences for the resulting signed material, which may sometimes omit phrases to keep up with the narrative, or interpreters would sign an utterance differently if they could give it a second thought.

A general property of SL interpretation (and hearing signers in general, as opposed to deaf signers) is that its linguistic structure tends to follow the structure imposed by the spoken language being translated (Janzen, 2005). This means that systems trained on such material may resemble hearing interpreters more than deaf translators.

9.4 Value created by this shared task

This shared task provides new insights and resources that previously did not exist for SLT, and that are valuable for the community.

We provided new training corpora and an official development and test set. We open-sourced a baseline system and code that is fully reproducible. We design protocols for human evaluation and adapt existing evaluation software accordingly. Lastly, the shared task resulted in the first openly available set of human judgements of automatic SL translations. Future work could use these scores for metric development, for instance.

10 Conclusion and future directions

In this paper we present the first WMT Shared Task on Sign Language Translation (WMT-SLT22). We consider automatic sign language translation, and sign language processing in general, to be of wide public interest and to have a high potential impact (§2).

Seven teams participated in this first edition of the shared task. Overall, we observed low system performance with an average human evaluation score of about 4 out of 100 (for the best-performing system), which is not usable in practice. The main reasons for this outcome are a lack of usable training data, a modality gap (considering that most existing work in MT is based on text) and a lack of basic NLP tools specifically for sign languages.

Future of the shared task Future iterations of the shared task could introduce more language pairs and larger training data. Since this year all submissions are signed-to-spoken systems, the shared task could also focus more on sign language generation going forward.

Furthermore, we will consider introducing additional MT-related tasks such as a sign language version of the metrics task. This perhaps requires a better distribution of human evaluation scores, as our current set of scores very much focuses on both ends of the score spectrum (we do not have many mid-range scores).

Finally, future human evaluation experiments for spoken-to-signed translation could be run differently than explained in this paper. Namely, for campaigns where a sign language is the target language the evaluation could be reference-based instead of source-based. The advantage of this change would be that deaf evaluators can perform this evaluation, instead of hearing interpreters for whom in this case the target language is not their first language.

11 Ethical statement

Within this shared task, two main ethical considerations emerge: the potential impact of SL technology on target users and privacy considerations.

Research in sign language processing, if not executed carefully, may inadvertently cause harm to end users, especially members of deaf communities. Hearing scientists should refrain from prescribing what sort of language technology should be accepted by deaf individuals and should avoid claiming that their approach "solves" any particular problem. Ideally, research of this nature should include deaf people, not only at evaluation time, but in the entire development cycle.

Secondly, there is a concern for the privacy of individuals depicted in SLP datasets. For the specific use case of sign language data, proper anonymisation is impossible since identifying details such as facial expressions are crucial for sign language communication. We have obtained written permission of all individuals shown in our datasets. Storing and processing pose estimation features instead of raw videos may be an alternative that provides anonymity (and has other generalization effects such as ignoring differences in race, gender, clothing, background etc.). However, in our shared task and related literature (Moryossef et al., 2021) video features outperform pose features.

Acknowledgments

This shared task was funded by the European Association for Machine Translation (EAMT) and by Microsoft AI for Accessibility. We are grateful for their support which enabled us to provide test data, human evaluation and interpretation in International Sign during the WMT conference.

The organizing committee further acknowledge funding from the following projects: the EU Horizon 2020 projects EASIER (grant agreement number 101016982) and SignON (101017255), the Swiss Innovation Agency (Innosuisse) flagship IICT (PFFS-21-47) and the German Ministry of Education and Research through the project SocialWear (01IW20002).

Thanks to Stanko Pavlica from FocusFilm and Robin Ribback from SWISSTXT for their help with data and licensing. Thanks to Tom Kocmi and Christian Federmann from Microsoft Munich for their help with the submission and evaluation platform. Thanks to Yvette Graham, Nitika Mathur and Tom Kocmi for providing expertise on statistical analysis.

Finally we would like to extend heartfelt thanks to the DSGS interpreters who performed our human evaluation: Michèle Berger, Sarah Caminada, Janine Criblez and Tanja Joseph.

References

Alan Agresti. 1996. *An introduction to categorical data analysis*, volume 135. Wiley New York.

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 Conference on Machine Translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Antonios Anastasopoulos, Loïc Barrault, Luisa Bentivogli, Marcely Zanon Boito, Ondřej Bojar, Roldano Cattoni, Anna Currey, Georgiana Dinu, Kevin Duh, Maha Elbayad, Clara Emmanuel, Yannick Estève, Marcello Federico, Christian Federmann, Souhir Gahbiche, Hongyu Gong, Roman Grundkiewicz, Barry Haddow, Benjamin Hsu, Dávid Javorský, Věra Kloudová, Surafel Lakew, Xutai Ma, Prashant Mathur, Paul McNamee, Kenton Murray, Maria Nădejde, Satoshi Nakamura, Matteo Negri, Jan Niehues, Xing Niu, John Ortega, Juan Pino, Elizabeth Salesky, Jiatong Shi, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Marco

Turchi, Yogesh Virkar, Alexander Waibel, Changhan Wang, and Shinji Watanabe. 2022. Findings of the IWSLT 2022 Evaluation Campaign. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 98–157, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Anonymous. 2022. Considerations for meaningful sign language machine translation based on glosses. Anonymous preprint under review.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-Scale Acquisition of Parallel Corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Claudia Bianchini and Fabrizio Borgia. 2012. Writing sign languages: analysis of the evolution of the signwriting system from 1995 to 2010, and proposals for future developments. In *Proceedings of the Intl Jubilee Congress of the Technical University of Varna*, pages 118–123.

Danielle Bragg, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudreault, Annelies Braffort, Naomi Caselli, Matt Huenerfauth, Hernisa Kacorri, Tessa Verhoef, Christian Vogler, and Meredith Ringel Morris. 2019. Sign Language Recognition, Generation, and Translation: An Interdisciplinary Perspective. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '19, pages 16–31, New York, NY, USA. Association for Computing Machinery.

Hannah Bull, Michèle Gouiffès, and Annelies Braffort. 2020. Automatic segmentation of sign language into subtitle-units. In *European Conference on Computer Vision*, pages 186–198. Springer.

Necati Cihan Camgöz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural Sign Language Translation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7784–7793.

Necati Cihan Camgöz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020a. Multi-channel transformers for multi-articulatory sign language translation. In *European Conference on Computer Vision*, pages 301–319.

Necati Cihan Camgöz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020b. Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10023–10033.

Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

João Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Niehues Jan, Stüker Sebastian, Sudoh Katsuitho, Yoshino Koichiro, and Federmann Christian. 2017. Overview of the IWSLT 2017 evaluation campaign. In *International Workshop on Spoken Language Translation*, pages 2–14.

Onno Crasborn. 2006. Nonmanual structures in sign language. *Encyclopedia of Language and Linguistics*, 8:668–672.

Mathieu De Coster, Dimitar Shterionov, Mieke Van Herreweghe, and Joni Dambre. 2022. Machine translation from signed to spoken languages: State of the art and challenges. *arXiv preprint arXiv:2202.03086*.

Mirella De Sisto, Dimitar Shterionov, Irene Murtagh, Myriam Vermeerbergen, and Lorraine Leeson. 2021. Defining Meaningful Units. Challenges in Sign Segmentation and Segment-Meaning Mapping (short paper). In *Proceedings of the 1st International Workshop on Automatic Translation for Signed and Spoken Languages (AT4SSL)*, pages 98–103, Virtual. Association for Machine Translation in the Americas.

Subhadeep Dey, Abhilash Pal, Cyrine Chaabani, and Oscar Koller. 2022. Clean Text and Full-Body Transformer: Microsoft's Submission to the WMT22 Shared Task on Sign Language Translation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Elisabeth Engberg-Pedersen. 1993. *Space in Danish Sign Language: The Semantics and Morphosyntax of the Use of Space in a Visual Language*. SIGNUM-Press.

Christian Federmann. 2018. Appraise Evaluation Framework for Machine Translation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 86–88, Santa Fe, New Mexico. Association for Computational Linguistics.

Michael Filhol. 2020. Elicitation and Corpus of Spontaneous Sign Language Discourse Representation Diagrams. In *Proceedings of the 9th Workshop on the Representation and Processing of Sign Languages at Language Resources and Evaluation Conference*, pages 53–60. European Language Resources Association (ELRA).

Joseph L. Fleiss. 1971. Measuring Nominal Scale Agreement Among Many Raters. *Psychological bulletin*, 76(5):378.

Jens Forster, Christoph Schmidt, Oscar Koller, Martin Bellgardt, and Hermann Ney. 2014. Extensions of the Sign Language Recognition and Translation Corpus RWTH-PHOENIX-Weather. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1911–1916, Reykjavik, Iceland. European Language Resources Association (ELRA).

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, Errors, and Context: A Large-Scale Study of Human Evaluation for Machine Translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous Measurement Scales in Human Evaluation of Machine Translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Yasser Hamidullah, Josef van Genabith, and Cristina España-Bonet. 2022. DFKI-MLT at WMT-SLT22: Spatio-temporal Sign Language Representation and Translation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Thomas Hanke. 2004. Hamnosys - representing sign language data in language resources and language processing contexts. In *LREC 2004, Workshop proceedings : Representation and processing of sign languages*, pages 1–6. Paris : ELRA.

Thomas Hanke, Susanne König, Reiner Konrad, Gabriele Langer, Patricia Barbeito Rey-Geißler, Dolly Blanck, Stefan Goldschmidt, Ilona Hofmann, Sung-Eun Hong, Olga Jeziorski, Thimo Kleyboldt, Lutz König, Silke Matthes, Rie Nishio, Christian Rathmann, Uta Salden, Sven Wagner, and Satu Worseck. 2020. MEINE DGS. Öffentliches Korpus der Deutschen Gebärdensprache, 3. Release.

Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. 2017. Learning spatio-temporal features with 3d residual networks for action recognition. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 3154–3160.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

Felix Hieber, Michael Denkowski, Tobias Domhan, Barbara Darques Barros, Celina Dong Ye, Xing Niu, Cuong Hoang, Ke Tran, Benjamin Hsu, Maria Nadejde, Surafel Lakew, Prashant Mathur, Anna Currey, and Marcello Federico. 2022. Sockeye 3: Fast Neural Machine Translation with PyTorch.

Lorenz Hufe and Eleftherios Avramidis. 2022. Experimental Machine Translation of the Swiss German Sign Language via 3D augmentation of body keypoints. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Terry Janzen. 2005. *Topics in signed language interpreting: Theory and practice*, volume 63. John Benjamins Publishing.

Trevor Johnston. 2010. From archive to corpus: Transcription and annotation in the creation of signed language corpora. *International Journal of Corpus Linguistics*, 15(1):106–131.

Trevor Johnston. 2011. Lexical Frequency in Sign Languages. *The Journal of Deaf Studies and Deaf Education*, 17(2):163–193.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Sang-Ki Ko, Chang Jo Kim, Hyedong Jung, and Choongsang Cho. 2019. Neural sign language translation based on human keypoint estimation. *Applied Sciences*, 9(13):2683.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To Ship or Not to Ship: An Extensive Evaluation of Automatic Metrics for Machine Translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Oscar Koller. 2020. Quantitative survey of the state of the art in sign language recognition. *arXiv preprint arXiv:2008.09918*.

Maria Kopf, Marc Schulder, and Thomas Hanke. 2021. Overview of Datasets for the Sign Languages of Europe.

Julia Kreutzer, Jasmijn Bastings, and Stefan Riezler. 2019. Joey NMT: A Minimalist NMT Toolkit for Novices. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 109–114, Hong Kong, China. Association for Computational Linguistics.

Taku Kudo. 2018. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

J R Landis and G G Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–174.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2018. Hallucinations in Neural Machine Translation.

Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. 2020. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1459–1469.

Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. 2019. MediaPipe: A Framework for Building Perception Pipelines. *CoRR*, abs/1906.08172.

Qingsong Ma, Yvette Graham, Timothy Baldwin, and Qun Liu. 2017. Further investigation into reference bias in monolingual evaluation of machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2466–2475, Copenhagen, Denmark. Association for Computational Linguistics.

Caio DD Monteiro, Christy Maria Mathew, Ricardo Gutierrez-Osuna, and Frank Shipman. 2016. Detecting and identifying sign languages through visual features. In *2016 IEEE International Symposium on Multimedia (ISM)*, pages 287–290. IEEE.

Sara Morrissey. 2011. Assessing three representation methods for sign language machine translation and evaluation. In *Proceedings of the 15th annual meeting of the European Association for Machine Translation (EAMT 2011), Leuven, Belgium*, pages 137–144.

Amit Moryossef and Yoav Goldberg. 2021. Sign Language Processing. `https://sign-language-processing.github.io/`.

Amit Moryossef and Mathias Müller. 2021a. pose-format: Library for viewing, augmenting, and handling .pose files. `https://github.com/AmitMY/pose-format`.

Amit Moryossef and Mathias Müller. 2021b. Sign Language Datasets. `https://github.com/sign-language-processing/datasets`.

Amit Moryossef, Ioannis Tsochantaridis, Joe Dinn, Necati Cihan Camgöz, Richard Bowden, Tao Jiang, Annette Rios, Mathias Müller, and Sarah Ebling. 2021. Evaluating the immediate applicability of pose estimation for sign language recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 10166v1. 2021 ChaLearn Looking at People Sign Language Recognition in the Wild Workshop at CVPR.

Mathias Müller, Annette Rios, and Amit Moryossef. 2022. Sockeye baseline models for sign language translation. `https://github.com/bricksdont/sign-sockeye-baselines`.

Ellen Ormel and Onno Crasborn. 2012. Prosodic correlates of sentences in signed languages: A literature review and suggestions for new types of studies. *Sign Language Studies*, 12(2):279–315.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Pamela Perniss, Asli Özyürek, and Gary Morgan. 2015. The influence of the visual modality on language structure and conventionalization: Insights from sign language and gesture. *Topics in Cognitive Science*, 7(1):2–11.

Elena Pizzuto and Paola Pietrandrea. 2001. The Notation of Signed Texts: Open Questions and Indications for Further Research. *Sign Language & Linguistics*, 4:29–45.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtskỳ. 2020. Transforming Machine Translation: a Deep Learning System Reaches News Translation Quality Comparable to Human Professionals. *Nature communications*, 11(1):1–15.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The Curious Case of Hallucinations in Neural Machine Translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1172–1183, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A Neural Framework for MT Evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ben Saunders, Necati Cihan Camgöz, and Richard Bowden. 2020a. Adversarial training for multi-channel sign language production. In *The 31st British Machine Vision Virtual Conference*. British Machine Vision Association.

Ben Saunders, Necati Cihan Camgöz, and Richard Bowden. 2020b. Everybody sign now: Translating spoken language to photo realistic sign language video. *arXiv preprint arXiv:2011.09846*.

Ben Saunders, Necati Cihan Camgöz, and Richard Bowden. 2020c. Progressive transformers for end-to-end sign language production. In *European Conference on Computer Vision*, pages 687–705.

Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. 2022. Signing at Scale: Learning to Co-Articulate Signs for Large-Scale Photo-Realistic Sign Language Production. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning Robust Metrics for Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich and Martin Volk. 2011. Iterative, MT-based Sentence Alignment of Parallel Texts. In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011)*, pages 175–182, Riga, Latvia. Northern European Association for Language Technology (NEALT).

Rico Sennrich and Biao Zhang. 2019. Revisiting Low-Resource Neural Machine Translation: A Case Study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

Bowen Shi, Diane Brentari, Gregory Shakhnarovich, and Karen Livescu. 2022. TTIC's WMT-SLT 22 Sign Language Translation System. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Anita Slonimska, Asli Özyürek, and Olga Capirci. 2021. Using Depiction for Efficient Communication in LIS (Italian Sign Language). *Language and Cognition*, 13(3):367–396.

Valerie Sutton. 1990. *Lessons in sign writing*. SignWriting.

Laia Tarrés, Gerard I. Gállego, Xavier Giró i Nieto, and Jordi Torres. 2022. Tackling Low-Resource Sign Language Translation: UPC at WMT-SLT 22. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Gül Varol, Liliane Momeni, Samuel Albanie, Triantafyllos Afouras, and Andrew Zisserman. 2021. Read and attend: Temporal localisation in sign language videos. In *CVPR*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Harry Walsh, Ben Saunders, and Richard Bowden. 2022. Changing the representation: Examining language representation for neural sign language production. In *LREC 2022*.

Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. 2020. SOLOv2: Dynamic and Fast Instance Segmentation. In *Advances in Neural Information Processing Systems*, volume 33, pages 17721–17732. Curran Associates, Inc.

Rosalee Wolfe, John C. McDonald, Thomas Hanke, Sarah Ebling, Davy Van Landuyt, Frankie Picron, Verena Krausneker, Eleni Efthimiou, Evita Fotinea, and Annelies Braffort. 2022. Sign language avatars: A question of representation. *Information*, 13(4):206.

Bencie Woll. 2013. 9091 The History of Sign Language Linguistics. In *The Oxford Handbook of the History of Linguistics*. Oxford University Press.

Kayo Yin, Amit Moryossef, Julie Hochgesang, Yoav Goldberg, and Malihe Alikhani. 2021. Including Signed Languages in Natural Language Processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7347–7360, Online. Association for Computational Linguistics.

Kayo Yin and Jesse Read. 2020. Better sign language translation with stmc-transformer. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5975–5989.

A Appraise instructions to human evaluators

A.1 Sign-to-text direction

A.1.1 English

Below you see a document with 10 sentences in Swiss-German Sign Language (Deutschschweizer Gebärdensprache (DSGS)) (left columns) and their corresponding candidate translations in German (Deutsch) (right columns). Score each candidate sentence translation in the document context. You may revisit already scored sentences and update their scores at any time by clicking on a source video.

Assess the translation quality on a continuous scale using the quality levels described as follows:

- 0: Nonsense/No meaning preserved: Nearly all information is lost between the translation and source. Grammar is irrelevant.
- 2: Some Meaning Preserved: The translation preserves some of the meaning of the source but misses significant parts. The narrative is hard to follow due to fundamental errors. Grammar may be poor.
- 4: Most Meaning Preserved and Few Grammar Mistakes: The translation retains most of the meaning of the source. It may have some grammar mistakes or minor contextual inconsistencies.
- 6: Perfect Meaning and Grammar: The meaning of the translation is completely consistent with the source and the surrounding context. The grammar is also correct.

Please score the overall document translation quality (you can score the whole document only after scoring all individual sentences first). Assess the translation quality on a continuous scale using the quality levels described as follows:

- 0: Nonsense/No meaning preserved: Nearly all information is lost between the translation and source. Grammar is irrelevant.
- 2: Some Meaning Preserved: The translation preserves some of the meaning of the source but misses significant parts. The narrative is hard to follow due to fundamental errors. Grammar may be poor.
- 4: Most Meaning Preserved and Few Grammar Mistakes: The translation retains most of the meaning of the source. It may have some grammar mistakes or minor contextual inconsistencies.
- 6: Perfect Meaning and Grammar: The meaning of the translation is completely consistent with the source and the surrounding context. The grammar is also correct.

A.1.2 German

Unten sehen Sie ein Dokument mit 10 Sätzen in Deutschschweizer Gebärdensprache (DSGS) (linke Spalten) und die entsprechenden möglichen Übersetzungen auf Deutsch (rechte Spalten). Bewerten Sie jede mögliche Übersetzung des Satzes im Kontext des Dokuments. Sie können bereits bewertete Sätze jederzeit durch Anklicken eines Eingabevideos erneut aufrufen und die Bewertung aktualisieren.

Bewerten Sie die Übersetzungsqualität auf einer kontinuierlichen Skala mit Hilfe der nachfolgend beschriebenen Qualitätsstufen:

- 0: Unsinn/Bedeutung nicht erhalten: Fast alle Informationen zwischen Übersetzung und Eingabevideo sind verloren gegangen. Die Grammatik ist irrelevant.
- 2: Ein Teil der Bedeutung ist erhalten: Die Übersetzung behält einen Teil der Bedeutung der Quelle bei, lässt aber wichtige Teile aus. Die Erzählung ist aufgrund von grundlegenden Fehlern schwer zu verstehen. Die Grammatik kann mangelhaft sein.

- 4: Der grösste Teil der Bedeutung ist erhalten und es gibt nur wenige Grammatikfehler: Die Übersetzung behält den grössten Teil der Bedeutung der Quelle bei. Sie kann einige Grammatikfehler oder kleinere kontextuelle Unstimmigkeiten aufweisen.
- 6: Perfekte Bedeutung und Grammatik: Die Bedeutung der Übersetzung stimmt vollständig mit der Quelle und dem umgebenden Kontext (falls zutreffend) überein. Auch die Grammatik ist korrekt.

Bitte bewerten Sie die Übersetzungsqualität des gesamten Dokuments. (Sie können das Dokument erst bewerten, nachdem Sie zuvor alle Sätze einzeln bewertet haben.) Bewerten Sie die Übersetzungsqualität auf einer kontinuierlichen Skala mit Hilfe der nachfolgend beschriebenen Qualitätsstufen:

- 0: Unsinn/Bedeutung nicht erhalten: Fast alle Informationen zwischen Übersetzung und Eingabevideo sind verloren gegangen. Die Grammatik ist irrelevant.
- 2: Ein Teil der Bedeutung ist erhalten: Die Übersetzung behält einen Teil der Bedeutung der Quelle bei, lässt aber wichtige Teile aus. Die Erzählung ist aufgrund von grundlegenden Fehlern schwer zu verstehen. Die Grammatik kann mangelhaft sein.
- 4: Der grösste Teil der Bedeutung ist erhalten und es gibt nur wenige Grammatikfehler: Die Übersetzung behält den grössten Teil der Bedeutung der Quelle bei. Sie kann einige Grammatikfehler oder kleinere kontextuelle Unstimmigkeiten aufweisen.
- 6: Perfekte Bedeutung und Grammatik: Die Bedeutung der Übersetzung stimmt vollständig mit der Quelle und dem umgebenden Kontext (falls zutreffend) überein. Auch die Grammatik ist korrekt.

A.2 Text-to-sign direction

A.2.1 English

Below you see a document with 10 sentences in German (Deutsch) (left columns) and their corresponding candidate translations in Swiss-German Sign Language (Deutschschweizer Gebärdensprache (DSGS)) (right columns). Score each candidate sentence translation in the document context. You may revisit already scored sentences and update their scores at any time by clicking at a source text.

Assess the translation quality on a continuous scale using the quality levels described as follows:

- 0: Nonsense/No meaning preserved: Nearly all information is lost between the translation and source. Naturalness of motion is irrelevant.
- 2: Some Meaning Preserved: The translation preserves some of the meaning of the source but misses significant parts. The narrative is hard to follow due to fundamental errors. Naturalness of motion may be poor.
- 4: Most Meaning Preserved and Acceptable Natural Motion: The translation retains most of the meaning of the source. It may have some minor mistakes or contextual inconsistencies. Motion may appear unnatural.
- 6: Perfect Meaning and Naturalness: The meaning of the translation is completely consistent with the source and the surrounding context. Motion is natural.

Please score the overall document translation quality (you can score the whole document only after scoring all individual sentences first). Assess the translation quality on a continuous scale using the quality levels described as follows:

- 0: Nonsense/No meaning preserved: Nearly all information is lost between the translation and source. Naturalness of motion is irrelevant.
- 2: Some Meaning Preserved: The translation preserves some of the meaning of the source but misses significant parts. The narrative is hard to follow due to fundamental errors. Naturalness of motion may be poor.

- 4: Most Meaning Preserved and Acceptable Natural Motion: The translation retains most of the meaning of the source. It may have some minor mistakes or contextual inconsistencies. Motion may appear unnatural.
- 6: Perfect Meaning and Naturalness: The meaning of the translation is completely consistent with the source. Motion is natural.

A.2.2 German

Unten sehen Sie ein Dokument mit 10 Sätzen auf Deutsch (linke Spalten) und die entsprechenden möglichen Übersetzungen in Deutschschweizer Gebärdensprache (DSGS) (rechte Spalten). Bewerten Sie jede mögliche Übersetzung des Satzes im Kontext des Dokuments. Sie können bereits bewertete Sätze jederzeit durch Anklicken eines Quelltextes erneut aufrufen und die Bewertung aktualisieren.

Bewerten Sie die Übersetzungsqualität auf einer kontinuierlichen Skala mit Hilfe der nachfolgend beschriebenen Qualitätsstufen:

- 0: Unsinn/Bedeutung nicht erhalten: Fast alle Informationen zwischen Übersetzung und Ausgangstext sind verloren gegangen. Es ist irrelevant, ob die Bewegungen natürlich sind.
- 2: Ein Teil der Bedeutung ist erhalten: Die Übersetzung behält einen Teil der Bedeutung der Quelle bei, lässt aber wichtige Teile aus. Die Erzählung ist aufgrund von grundlegenden Fehlern schwer zu verstehen. Bewegungen können mangelhaft sein.
- 4: Der grösste Teil der Bedeutung ist erhalten und die Bewegungen sind akzeptabel: Die Übersetzung behält den grössten Teil der Bedeutung der Quelle bei. Sie kann kleine Fehler oder kleinere kontextuelle Unstimmigkeiten aufweisen. Bewegungen sehen teilweise nicht natürlich aus.
- 6: Perfekte Bedeutung und Natürlichkeit: Die Bedeutung der Übersetzung stimmt vollständig mit der Quelle und dem umgebenden Kontext (falls zutreffend) überein. Bewegungen wirken natürlich.

Bitte bewerten Sie die Übersetzungsqualität des gesamten Dokuments. (Sie können das Dokument erst bewerten, nachdem Sie zuvor alle Sätze einzeln bewertet haben.) Bewerten Sie die Übersetzungsqualität auf einer kontinuierlichen Skala mit Hilfe der nachfolgend beschriebenen Qualitätsstufen:

- 0: Unsinn/Bedeutung nicht erhalten: Fast alle Informationen zwischen Übersetzung und Ausgangstext sind verloren gegangen. Es ist irrelevant, ob die Bewegungen natürlich sind.
- 2: Ein Teil der Bedeutung ist erhalten: Die Übersetzung behält einen Teil der Bedeutung der Quelle bei, lässt aber wichtige Teile aus. Die Erzählung ist aufgrund von grundlegenden Fehlern schwer zu verstehen. Bewegungen können mangelhaft sein.
- 4: Der grösste Teil der Bedeutung ist erhalten und die Bewegungen sind akzeptabel: Die Übersetzung behält den grössten Teil der Bedeutung der Quelle bei. Sie kann kleine Fehler oder kleinere kontextuelle Unstimmigkeiten aufweisen. Bewegungen sehen teilweise nicht natürlich aus.
- 6: Perfekte Bedeutung und Natürlichkeit: Die Bedeutung der Übersetzung stimmt vollständig mit der Quelle und dem umgebenden Kontext (falls zutreffend) überein. Bewegungen wirken natürlich.

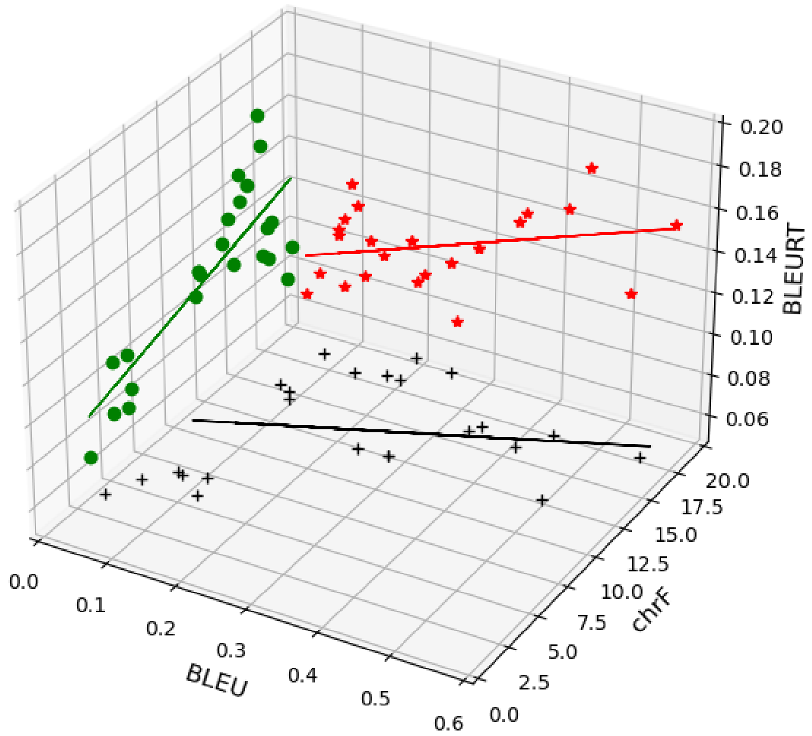


Figure 6: Correlation between the metrics used in the automatic evaluation. Automatic evaluation scores projected into the 2D spaces for BLEU–chrF (black crosses, $r = 0.447$), BLEU–BLEURT (red stars, $r = 0.703$) and chrF–BLEURT (green dots, $r = 0.443$).

B Correlation between automatic metrics

Metrics do not correlate well with each other, especially if chrF is compared to a second metric. Figure 6 plots the projection for the scores on the full test set by metric pair for the 23 submissions and the baseline. The Pearson correlation shows that metrics are far from a linear relation: BLEU–chrF has $r = 0.447$, BLEU–BLEURT $r = 0.703$ and chrF–BLEURT $r = 0.443$. Spearman correlation, accounting only for monotonicity, is lower in the three cases specially for chrF–BLEURT ($\rho = 0.259$), with $\rho = 0.421$ for BLEU–chrF and $\rho = 0.633$ for BLEU–BLEURT.

C Feedback from evaluators

Table 8 lists detailed by evaluators regarding the human evaluation procedure and the Appraise system. Two out of four evaluators submitted a response.

	Answer 1	Answer 2
What is your experience in assessing machine translation outputs?		
	None: this was my first time	Low: I have done it once or a long time ago
Please specify how much you agree or disagree with the following statements.		
Generally, my experience with the tool was positive	Agree	Strongly agree
Instructions were clear	Agree	Strongly agree
Quality levels 0-6 were helpful to me	Agree	Strongly Agree
Source videos/texts were understandable	Neutral	Strongly Agree
There was too much repetitiveness	Disagree	Agree
Documents were too long	Disagree	Strongly Disagree
Segments were too short	Neutral	Disagree
In some cases, the context was insufficient	Strongly Agree	Disagree
I experienced technical issues	Agree	Agree
I would be willing to do similar work in future	Strongly agree	Agree
Please provide more details related to the statements above that you think can be useful to us. What was most troublesome? What could we improve?		
	it would be very helpful, if the video started automatically when moving to the next segment. (some did, but many more did not) It would save a click. Also, the submit button could be on the left side under the 0 score (at the moment, as most translation are not yet good quality)	-
What were the main or most common issues with the automatic translations?		
	This question is not clear to me. You mean on a technical level or something else? meaning was garbage, some did not know the German Umlaute äüö	-
This evaluation campaign featured the Direct Assessment with Scalar Quality Metrics method. What do you think about this method? On a scale between -3 (negative) and 3 (positive) it was...		
difficult/easy	2	2
stressful/relaxed	2	-2
laborious/effortless	1	2
slow/fast	0	0
inefficient/efficient	2	2
boring/exciting	-2	3
complicated/simple	2	2
annoying/enjoyable	0	1
limiting/creative	-2	0
impractical/practical	2	0

Table 8: Feedback from evaluators about the human evaluation setup and the Appraise platform.

D Human evaluation score distribution

To complement our analysis we show the distribution of scores for each system in Figure 7. The set of scores (excluding zero scores, which are not shown in the figure) resembles a bimodal distribution, with most of the scores residing at both ends of the spectrum. MSMUNICH is the system with the most scores in the highest-quality bucket.

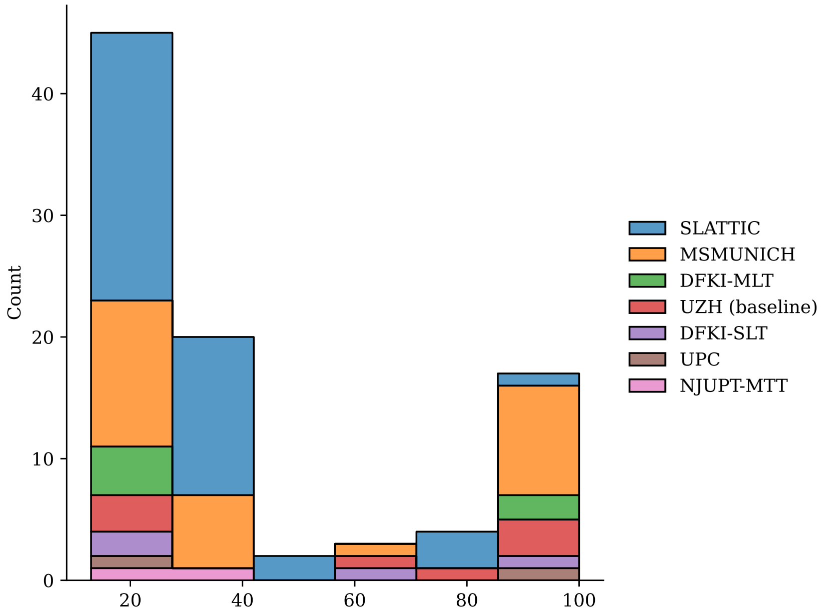


Figure 7: Distribution of human evaluation scores for all submitted systems discretized in seven bins, excluding scores of bin 0 (lowest quality).

Findings of the WMT’22 Shared Task on Large-Scale Machine Translation Evaluation for African Languages

David Ifeoluwa Adelani[♠] **Md Mahfuz Ibn Alam**[†] **Antonios Anastasopoulos**[†] **Akshita Bhagia**[◊]
Marta R. Costa-jussà[♡] **Jesse Dodge**[◊] **Fahim Faisal**[†] **Natalia Fedorova**[‡] **Christian Federmann**[⊗]
Francisco Guzmán[♡] **Sergey Koshelev**[‡] **Jean Maillard**[♡] **Vukosi Marivate**[#] **Jonathan Mbuya**[†]
Alexandre Mourachko[♡] **Safiyyah Saleem**[♡] **Holger Schwenk**[♡] **Guillaume Wenzek**[♡]
[†]George Mason University [♠]University College London [♡]Meta AI
[⊗]Microsoft Research [‡]Toloka [#]University of Pretoria [◊]AI2

Abstract

We present the results of the WMT’22 Shared Task on Large-Scale Machine Translation Evaluation for African Languages. The shared task included both a data and a systems track, along with additional innovations, such as a focus on African languages and extensive human evaluation of submitted systems. We received 14 system submissions from 8 teams, as well as 6 data track contributions. We report a large progress in the quality of translation for African languages since the last iteration of this shared task: there is an increase of about 7.5 BLEU points across 72 language pairs, and the average BLEU scores went from 15.09 to 22.60.

1 Introduction

A large portion of the world’s population speak *low-resource languages*, and would benefit from improvements in translation quality on their native languages. Recent advances in translation quality, particularly from massively multilingual models (Fan et al.; Ma et al., 2021), have enabled progress in the translation quality of low-resource languages. However, many languages have seen little to no progress. This is particularly true for African languages. For example, in the 2021 Large Scale Multilingual Evaluation shared task (Wenzek et al., 2021), there was a progress[2] of +19.3 avg. BLEU when translating into languages like Irish and Welsh (included in the Other Indo-European grouping), but only a progress of +3.5 avg. BLEU when translating into languages like Fula and Igbo (included in the Nilotic/Atlantic Congo grouping).

The African continent is home to a rich diversity of languages. Around a third of the world’s living languages are from Africa and only a small fraction of the resources in NLP and Machine translation are dedicated to them (Orife et al., 2020). As a result, African language speakers do not benefit from language technologies similarly to other Global North language speakers (Blasi et al., 2022). A major (but not the sole) hurdle to building language technologies for African languages is data availability (Joshi et al., 2021), with significant efforts underway stemming –importantly– from Africa itself (Nekoto et al., 2020).

To bring attention of the research community to the challenges of translating African Languages, this year we focus on the multilingual evaluation of 24 African Languages together with French and English. We base our evaluation on the benchmark provided by FLORES (Goyal et al., 2022), and its recent expansion to more African languages (NLLB Team et al., 2022).

In this second multilingual large-scale shared task, we evaluate the progress on massively multilingual translation for African Languages in a non-English-centric way. We propose 100 evaluation language pairs, based on regional clusters (South/South-East Africa, Horn of Africa and Central/East Africa, Nigeria and Gulf of Guinea, and Central Africa; and pivot languages (English, French). This year is characterized by two further innovations: first, a data track is included, in which participants share corpora to be used during the shared task; second, we perform human evaluation for a subset of the language pairs.

In the remainder of this paper, we describe the task setup, the participants, and the official results for the task. We also analyze the results to understand better the languages for which progress has been attained, and those where a gap in quality is still observed. Finally, we propose future directions for other tasks in the future.

Author names are sorted in alphabetical order.

[2] Progress was determined by comparing the average BLEU score between the best system in the task vs. the baseline.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 773 - 800
December 7-8, 2022 ©2022 Association for Computational Linguistics

2 Shared Task and Tracks

This year's task focuses on the 24 African languages listed in Table 1, along with French and English which are included as colonial linguae francae for evaluation purposes. These languages are all supported by the FLORES benchmark in its most recent expansion (NLLB Team et al., 2022).

Afrikaans	afr	Oromo	orm
Amharic	amh	Shona	sna
Chichewa	nya	Somali	som
Nigerian Fulfulde	fuv	Swahili	swh
Hausa	hau	Swati	ssw
Igbo	ibo	Tswana	tsn
Kamba	kam	Umbundu	umb
Kinyarwanda	kin	Wolof	wol
Lingala	lin	Xhosa	xho
Luganda	lug	Xitsonga	tso
Luo	luo	Yorùbá	yor
Northern Sotho	nso	Zulu	zul

Table 1: Focus African languages for this shared task. In addition to these, we also include French and English.

The human and automatic evaluation is based around 100 language directions, selected based on translator and annotator availability:

- **Languages of South and Southeast Africa**: xho-zul, zul-sna, sna-afr, afr-ssw, ssw-tsn, tsn-tso, tso-nso, nso-xho (8 directions).
- **Languages of the Horn of Africa**: swh-amh, amh-swh, luo-orm, som-amh, orm-som, swh-luo, amh-luo, luo-som (8 directions).
- **Languages of West Africa**: hau-ibo, ibo-yor, yor-fuv, fuv-hau, ibo-hau, yor-ibo, fuv-yor, hau-fuv, wol-hau, hau-wol, fuv-wol, wol-fuv (12 directions).
- **Languages of Central Africa**: kin-swh, lug-lin, nya-kin, swh-lug, lin-nya, lin-kin, kin-lug, nya-swh (8 directions).
- **Cross-regional pairs**: amh-zul, yor-swh, swh-yor, zul-amh, kin-hau, hau-kin, nya-som, som-nya, xho-lug, lug-xho, wol-swh, swh-wol (12 directions)
- **English pivots:** 22 languages translated into and from eng: afr, amh, nya, fuv, hau, ibo, kam, kin, lug, luo, nso, orm, sna, som, swh, ssw, tsn, umb, xho, tso, yor, zul (44 directions).
- **French pivots:** 4 languages translated into and from fra: kin, lin, swh, wol (8 directions).

3 Data Track

The data track focused on the contribution of novel corpora. Participants were welcomed to open-source and share monolingual, bilingual or multilingual datasets relevant to the training of MT models for this year's set of languages. There were seven submissions in this track:

LAVA[3] LAVA Corpus contains millions of parallel bilingual sentences, which are mined from Common Crawl. It covers five African languages.

MAFAND-MT (Adelani et al., 2022)[4] contains a few thousand high-quality and human translated parallel sentences for 21 African languages in the **news** domain. Each language has between 1,400 - 34,500 parallel sentences for training and/or evaluation. The languages covered are Amharic, Bambara, Ghomala, Ewe, Fon, Hausa, Igbo, Kinyarwanda, Luganda, Dholuo, Mossi, Chichewa, Nigerian-Pidgin, chiShona, Swahili, Setswana, Twi, Wolof, Yorùbá, isiXhosa, and isiZulu. These languages include 7 languages which were not present in the Shared task (and which are not reported therefore in Table 2).

KenTrans[5] This project produced a parallel corpus between Swahili and 2 other Kenya Languages: Dholuo and Luhya. The Luhya Language has several dialects. In the project 3 dialects were chosen as a start: Lumarachi, Logooli and Lubukusi. A total of 12,400 sentences were translated to Kiswahili from a sample of Dholuo and Luhya (1500 Dholuo-Kiswahili sentence pairs and 10,900 Luhya-Kiswahili sentence pairs). This corpus has an extension to speech in the version of Kencorpus (Wanjawa et al., 2022)[6].

Monolingual African languages from ParaCrawl[7] This release contains derived corpora built from language classified extracts of the ParaCrawl project. Monolingual data in this release comes from the Internet Archives and targeted crawls performed in the paracrawl project with document level language classification.

[3] https://drive.google.com/drive/folders/\179AkJ0P3fZMFS0rIyEBBDZ-WICs2wpWU

[4] https://github.com/masakhane-io/lafand-mt

[5] https://doi.org/10.7910/DVN/NOAT0W

[6] https://doi.org/10.7910/DVN/6N5V1K

[7] https://data.statmt.org/martin/

Dataset	African languages covered	No. of sentences	Participating Teams
LAVA	afr, kin, lug, nya, swa	3,225,801	Bytedance, GMU, ANVITA
MAFAND-MT	amh, hau, ibo, kin, lug, luo, nya, sna, swh, tsn, wol, xho, yor, zul	102,135	Bytedance, Tencent, DENTRA, ANVITA, Masakhane, GMU
KenTrans	luy, luo, swa	12,400	Bytedance, Tencent, DENTRA, ANVITA
ParaCrawl	afr, amh, fuv, hau, ibo, kam, lin, lug, luo, nso, nya, orm, sna, ssw, swh, tsn, tso, umb, wol, xho, yor, zul	22,349,179	Bytedance, Tencent, DENTRA, GMU
SA corpus	nso, tsn, xho, zul	160,035	Bytedance, Tencent, DENTRA, ANVITA
WebCrawlAfrican	afr, ling, ssw, amh, lug, tsn, nya, hau, orm, xho, ibo, tso, yor, swh, zul	695,000	Bytedance, Tencent, DENTRA, ANVITA

Table 2: Dataset submissions: covered Shared task languages, dataset sizes (i.e. number of parallel sentences in all translation directions), and participating teams that made use of the dataset in their submission.

SA Languages[8] The dataset was constructed using public available data mostly from South African Government websites.

WebCrawlAfrican (Vegi et al., 2022b) [9] Web Crawl African is a collection of African Multilingual parallel corpora comprising of 695,000 (approx) sentence pairs, covering 15 African languages plus English and 73 language pairs. African languages covered include Afrikaans, Lingala, Swati, Amharic, Luganda, Tswana, Chichewa, Hausa, Oroma, Xhosa, Igbo, Xitsonga, Yoruba, Swahili, Zulu. It covers variety of domains political, stories, religious and songs. Corpora have sentences covering both formal and informal writing styles.

This participation is summarised in Table 2 which includes language covered, dataset size and participating teams that currently used the dataset for their submission. All datasets where at least used by 3 teams in the evaluation. Note that the most used corpus was MAFAND-MT which was used by 6 different teams.

4 System Track

We provided a selection of training corpora, i.e. parallel sentences, to enable training of the MT systems. Submissions in the constrained translation track were only allowed to use data from the following sources:

- all corpora from the data track (see section 3);
- parallel corpora from OPUS (Tiedemann, 2012);[10]
- parallel corpora mined from Common Crawl using the LASER3 multilingual sentence encoder.

Participants who used other resources, had to submit to the unconstrained translation track.

Publicly available resources for African languages are very limited, for some of them less than fifty thousand sentences of bitexts are available. In addition to human translated sentences, several approaches were proposed to automatically mine parallel sentence from large collections of monolingual data. Unfortunately, recent approaches like ParaCrawl or CCMatrix (Schwenk et al., 2021) cover only few African languages. We extended the basic idea of mining based on a similarity measure in an multilingual embedding space (Artetxe and Schwenk, 2019) and developed sentence encoders for all African languages of this evaluation. We then performed bitext mining against 21.5 billion English sentences from Common Crawl, and 3.3 million sentences in French, respectively. These resources as well as the sentence encoders were made available to the participants of this evaluation. A detailed description of this mining approach can be found in Heffernan et al. (2022).

4.1 Participating Teams

CAIR ANVITA (Vegi et al., 2022a). The ANVITA-1.0 MT system is an English-centric multilingual transformer model for 24 African languages. The authors applied several heuristics to filter the data released for the shared task. They showed that that using larger Transformer model (24 encoder, 6 decoder) performs much better than smaller Transformer model (6 encoder, 6 decoder). Furthermore, they obtained some improvements by using an ensemble of last two epochs of Deep Transformer (24 encoder, 6 decoder).

[8] https://drive.google.com/drive/folders\/1jYwpxEdRxqXlB7BSmE6JxDar61U91xfI

[9] https://github.com/pavanpankaj/Web-Crawl-African

[10] https://opus.nlpl.eu/

Cape Town (Elmadani et al., 2022). This system was focused on eight South and South-East African languages. The authors trained a multilingual NMT system (foreign to English and English to foreign, with some non English directions). The authors focused on exploring the best sub-word representation (BPE vs overlap-based BPE) and its effects on downstream performance for low-resource languages. Further, the authors explore creating synthetic data through back-translation and explore sampling techniques to balance the corpora.

GMU (Ibn Alam and Anastasopoulos, 2022). This system was based on on fine-tuning pretrained multilingual DeltaLM on 26 languages (625 translation directions). The fine-tuning was based on language- and language-family- (phylogeny) inspired adapter units (Faisal and Anastasopoulos, 2022) to improve its performance for African languages. The results show that a language-adapter-based fine-tuning significantly out-performs direct fine-tuning, but making use of family/sub-family adapters only helps in a few cases.

IIAI DenTra (Kamboj et al., 2022). This multilingual model combines a traditional translation loss with self-supervised tasks that can make use of unlabeled monolingual data. The resulting model performs denoising tasks (shuffling, masking) in conjunction with both translation and backtranslation. It then fine-tunes the model to in-domain data and covers 24 languages.

Masakhane (Abdulmumin et al., 2022). This model is based on M2M-100 which is fine-tuned on training data hat has been cleaned by an auxiliary language models. The pre-trained language models were fine-tuned on *positive* samples (clean data) vs. *negative* samples coming from automatically aligned data. The authors find significant improvements from using the filtered data.

Tencent Borderline (Jiao et al., 2022). The borderline model is a large transformer model (1.02B params.) which is augmented with data for zero-shot pairs through tagged back-translation and self-translation. In addition, it uses distributionally robust optimization (DRO) to alleviate the data imbalance. Finally it also uses family language information to group target languages and finetune separate models for each group.

Bytedance VolcTrans (Qian et al., 2022). This is an unconstrained system that uses different sources of parallel data (constrained data, NLLB, self-procured data) and monolingual data (e.g. VOA news, Wikipedia). This model is also trained on data cleaned through a rich set of heuristic rules to prevent punctuation mismatches, overly short/long sentences, among others; together with an approach based on minimum description length (MDL) that removes noisy sentences. The data is augmented with back-translation coming from pivot languages (Eng/Fra). The model is trained with target language tags added to both the encoder and decoder inputs. Finally, it includes post-processing rules for Yoruba accents.

SRPH-DAI (Cruz and Sutawika, 2022). This model is based on mT5, with additional adapters fine-tuned to each translation task, and then merged using adapter fusion to perform task-composition. The model is trained over data that is filtered using a set of heuristics. This model doesn't use other data-augmentation techniques (e.g. BT).

4.2 Automatic Evaluation

We follow last year's shared task in relying in spBLEU (Goyal et al., 2022) due to the well-known limitations of traditional BLEU. In particular we use a sentencepiece (Kudo and Richardson, 2018) tokenizer trained on all FLORES-200, in the hope of producing a *universal* tokenizer that can adequately handle all languages we are dealing with. Finally, to compute BLEU, we apply SPM tokenization to the system output and the reference, and then calculate BLEU at the sentence-piece level. For readability, in this paper we use BLEU and spBLEU interchangeably.

We additionally report chrF++ (Popović, 2017), another metric relying on character n-gram F-score (chrF) alongside word-level unigram and bigram F-score. This metric has been shown to correlate particularly well with human judgments for languages with rich morphology.

For all results we rely on statistical significance tests, using paired bootstrap resampling (Koehn, 2004) with 1000 samples. We first rank all the systems with spBLEU then we take the highest-scored system as a baseline and compare it with systems that are below its rank. If the p-value is greater than 0.05 we bundle those systems together. If for a system the p-value is less than 0.05 that means we have found a statistically significantly worse system and we make that system the new baseline system and continue to go on.

4.3 Human Evaluation

A fixed sample of the outputs of the primary submissions was grouped by the language pairs and split into tasks for evaluation by crowd human annotators. Each task comprised the source sentence and two translations of the source sentence that were to be scored from 0 to 100 indicating the general quality of the translations.

The annotation was conducted via crowdsourcing on the Toloka platform [11] where a crowd labeling project was set up for each language pair.

Guidelines for Evaluation The guidelines for the scale were roughly based on the theory of levels of translation equivalence by Komissarov (1990) and were simplified in order to facilitate their usage by annotators without linguistic background . The crowd annotators were asked to evaluate the translations on a 0..100 continuous rating scale (Graham et al., 2013) where 0 was considered a very bad and 100 — a very good translation (see Figure 1). The scale represents a combined approach that requires the annotators to give each translation one score assessing both accuracy and fluency at the same time and taking into account factors like grammaticality, naturalness, conveying the same meaning, having the same communicative goal, representing the same situation, having the same style and preserving as many shades of meaning of the source sentence as possible. The annotators were asked to evaluate the sentences as a whole and not word by word, pay attention to unmotivated additions of omissions, grammatical mistakes and untranslated parts, check if set expressions and metaphors were translated according to the expressions used in the target language and use their feel for the language in general.

Annotator Selection We ran the crowd annotation projects in Toloka and used the following criteria for annotator selection:

- Location: Africa according to the evaluator's IP address
- Both languages of the respective pair should have been among the list of languages known by the annotator as they had specified in their Profile
- Wherever possible, we showed the projects only to those annotators who had passed the respective language tests built in Toloka. The platform asks the annotators to verify their language skills via tests to get access to tasks with this prerequisite and it had such tests for English and French.

Potential annotators did not know about the requirements and did not see the projects if they did not meet the criteria. This approach prevented them from deliberate manipulation of their user settings to get access to the projects.

Quality Control The annotators whose profile matched our requirements, were shown the guidelines. The guidelines were translated into the target language of the respective pair as another way of testing the annotator's language skills.

Then the annotators were given an exam to test their ability to complete the task. The exam consisted of five tasks structurally identical to the main ones (see Figure 1). Thus each task consisted of the source sentence, two translations and a continuous scale for scoring. The exams were generated automatically from the list of source sentences paired with target sentences. The target sentences for the exam comprised:

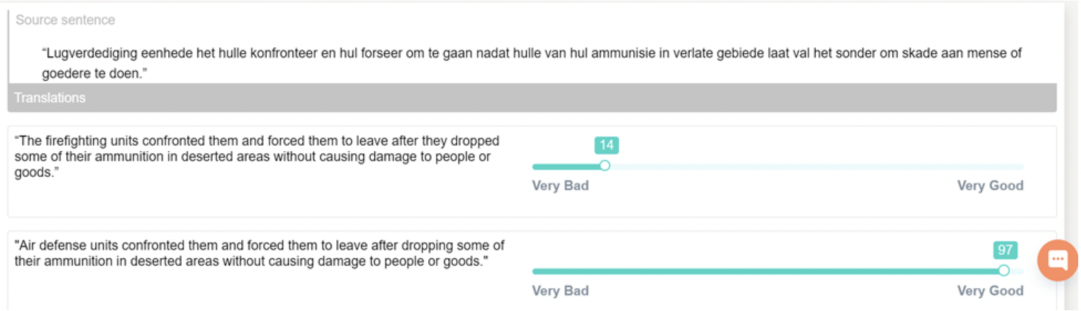
- Reference translations
- Reference translations with randomized word order (while still having correct capitalization when needed and ending with the respective punctuation mark)
- "Gibberish" sentences. Being potentially unknown to virtually any annotator, Sumerian was chosen as the source of "gibberish" control sentences [12] for eliminating cheaters. Furthermore, being unrelated to the languages in the dataset, it didn't confuse diligent annotators in terms of what pair of languages needs to be evaluated.

Each translation was assigned a golden score to be used as a control answer (Chida et al., 2022). Since the evaluation scale used is subjective and can show high variance, control intervals were introduced with the matching interval scores (score 0–33 = 0, 34–66 = 50, 66+ = 100). Randomized sentences as well as sentences in Sumerian were given golden scores 0 and reference translations were given a 100. Thus, if the annotator scored a bad sentence in a range of 0-33 it meant that they gave a correct answer and so on.

The task was accepted if both translations were given correct scores and rejected otherwise. The accuracy of the performer in an exam was calcu-

[11] https://toloka.ai

[12] https://etcsl.orinst.ox.ac.uk/cgi-bin/etcsl.cgi?text=c.1.8.1.1


Source sentence

"Lugverdediging eenhede het hulle konfronteer en hul forseer om te gaan nadat hulle van hul ammunisie in verlate gebiede laat val het sonder om skade aan mense of goedere te doen."

Translations

"The firefighting units confronted them and forced them to leave after they dropped some of their ammunition in deserted areas without causing damage to people or goods."

14

Very Bad Very Good

"Air defense units confronted them and forced them to leave after dropping some of their ammunition in deserted areas without causing damage to people or goods."

97

Very Bad Very Good

Figure 1: Screenshot of the interface with an annotated task comprising the source sentence and two translations randomly chosen from the outputs of the submitted models for the respective language pair (Afrikaans – English in the screenshot).

lated as the rate of accepted tasks among the five comprising the examination set. Exams where one of the languages (English or French) had been verified by the platform test required a 100% accuracy of completion (5/5 accepted tasks). For other languages the accuracy threshold was 80% to neutralize potential variance introduced by the automated creation of exams. These accuracy scores given in the exam were used as filters for the main pools of tasks, giving access to the main labeling only to the annotators who had passed the exam.

Another way of eliminating potential low-performers used both during the exam and the main set of tasks was banning the annotators for very fast responses. If an annotator submitted a page of five tasks within twenty seconds or less two times or more, they were banned from working on the project.

Score Normalization We convert the raw human scores to Z-scores:

$$z = \frac{x - \mu}{\sigma}$$

Note that we perform this operation at the annotator level; that means we compute the mean μ and standard deviation σ separately for each annotator and apply the transformation only on their scores.

Language Pairs Due to annotator availability, we only perform human evaluations in a smaller subset of our 100 language pairs. Hence the human score averages presented in all results only reflect averages for this subset, not all language pairs. The list of pairs is available in Table 15 in the Appendix.

5 Results

In this section we analyze both the human scores and automatic metrics for all the participants. We present the results and official ranking for the main task; the analysis of the performance from/to English, an to/from African languages; and the progress in performance w.r.t. to the previous year's task.

5.1 Main Results

The average results across the 100 evaluation language pairs are reported in Table 4. Note that we primarily only rank the *constrained* systems that were able to handle all language pairs.

System	# of pairs
Unconstrained	
Bytedance$_p$	52
Bytedance$_c$	48
Constrained	
Tencent$_p$	39
Tencent$_c$	26
GMU$_p$	17
DENTRA$_p$	9
GMU$_c$	7
Masakhane$_c$	1
SPRH-DAI$_c$	1
ANVITA, Masakhane$_p$ CapeTown, SPRH-DAI$_p$	0

Table 3: Number of evaluation language pairs (out of 100 total) that a system ranks (or ties for) best performance (spBLEU).

The best-performing constrained system on average is the Borderline (Tencent) system, followed by the GMU system with a little over 1 BLEU point difference between them. The only unconstrained submission is Bytedance's Volctrans, outperforming other systems by a significant margin of more than 4 BLEU points, but note however that it uses

Rank	System	Human	BLEU	spBLEU	chrF2
Systems handling all language pairs					
Constrained					
1	Tencent$_p$	0.39 ± 0.15	14.1 ± 1.1	17.6 ± 1.2	37.4 ± 1.3
	Tencent$_c$		14.0 ± 1.0	17.5 ± 1.2	37.2 ± 1.4
	GMU$_c$		13.3 ± 1.1	16.2 ± 1.1	35.4 ± 1.4
2	GMU$_p$	0.16 ± 0.28	13.3 ± 1.1	16.2 ± 1.1	35.4 ± 1.4
3	DENTRA$_p$	0.02 ± 0.51	10.4 ± 1.1	12.7 ± 1.2	30.5 ± 1.5
	SPRH-DAI$_c$		1.6 ± 1.2	2.0 ± 1.5	11.5 ± 0.4
4	SPRH-DAI$_p$	-1.4 ± 0.24	1.5 ± 1.6	1.8 ± 1.8	10.4 ± 0.5
Unconstrained					
U	ByteDance$_p$	0.53 ± 0.19	17.3 ± 1.2	21.9 ± 1.1	41.9 ± 1.2
U	ByteDance$_c$		17.3 ± 1.2	21.8 ± 1.1	41.7 ± 1.2
Systems handling partial language pairs					
P	ANVITA$_p$	0.24 ± 0.19	24.3 ± 1.1	26.3 ± 1.2	44.9 ± 1.2
P	ANVITA$_c$		23.8 ± 1.1	25.9 ± 1.2	44.5 ± 1.2
P	Capetown$_p$	0.09 ± 0.24	17.4 ± 1.0	21.3 ± 0.8	43.7 ± 0.7
P	Masakhane$_p$	0.05 ± 0.13	16.6 ± 1.0	19.0 ± 1.0	39.3 ± 1.1
P	Masakhane$_c$		11.9 ± 0.7	14.1 ± 0.7	35.0 ± 0.8

Table 4: Average results (presented here: mean ± standard error across all 100 evaluation language pairs), sorted by spBLEU. We only ranked constrained systems that handle all language pairs. We denote *unconstrained* submissions with U, and systems covering partial language pairs with P.

unconstrained data. All metrics, including human evaluation, result in a similar ranking among the 4 systems handling all language pairs.[13]

Table 3 lists the number of evaluation pairs for which each system ranks or ties for best performance. Among the constrained systems, the Tencent ones rank first for 39 language pairs, with the GMU one following with 17, and DENTRA with 9. The Masakhane and SPRH-DAI contrastive systems also rank first in one language pair each.

5.2 Performance by Language

Table 5 presents the best and worst performing language pairs. As shown, some language pairs end up with very good systems (e.g. Afrikaans, Swahili, Northern Sotho – at least when pairing with English). Wolof, Umbundu, and Kamba are the languages that most often show up as targets in the language pairs that systems struggle in, with BLEU scores below 10.

Results by for each individual language pair can be found in the Appendix Tables 16–115.

Source	Target	spBLEU
eng	umb	4.1
eng	kam	5.9
fra	wol	8.3
swh	wol	8.5
hau	wol	8.6
afr	eng	60.1
swh	eng	49.0
eng	afr	46.0
nso	eng	43.5
eng	swh	42.0

Table 5: Top-5 worst (top) and best (bottom) language pairs (result from best system).

5.3 Performance by Groups

Table 6 presents the best performance across all systems, averaged over the different language groups we defined above. Note that having colonial languages (English and French) as the target perhaps unsurprisingly leads to generally high performance (average more than 33 BLEU). Languages from South Africa seem to be easier to translate (average ∼29 BLEU) while languages from West Africa, and especially Wolof, lead to worse performance as targets (average ∼15 BLEU).

5.4 Progress from Last Year

To assess the progress made in African languages, we compare this year's best results with DeltaLM (Yang et al., 2021), the best system from last year's shared task. Note that this analysis only includes 72 of our 100 evaluation pairs, as some of the languages (e.g. Kinyarwanda or Swati) were

[13] For a fair comparison, Appendix Tables 13 and 14 present average scores for the partial language pairs that the Masakhane and CapeTown systems handle.

Group	as src	as trg	as both
South	29.4	22.8	19.9
Horn	21.9	20.0	16.5
West	20.3	15.1	14.4
Central	21.7	20.0	16.2
Colonial	22.3	33.7	N/A

Table 6: Average spBLEU of the best performing system summarized per language group.

added this year. In addition, note that last year's systems were scored using a different sentencepiece tokenizer than this year's ones.[14]

Across all 72 pairs, the average improvement is around 7.5 BLEU points. For around 93% of the 72 language pairs (67 pairs), there are improvements over last year's best system. We show the top-10 improved language pairs in Table 7. Most of the English-centric language pairs improve significantly, but african-to-african pairs benefit too, like the Swahili to Dholuo (swh-luo) pair which shows more than 12 BLEU points improvement. When indeed improving, the average improvement is more than 8 BLEU points (max: 33.5, min: 0.94).

Source	Target	2021	2022	Δ
nso	eng	9.9	43.5	33.5
eng	nso	9.5	30.5	21.1
yor	eng	7.1	25.1	18.0
hau	eng	22.2	40.0	17.8
orm	eng	10.6	28.3	17.7
eng	hau	16.4	31.5	15.1
som	eng	20.8	35.0	14.2
eng	luo	3.5	16.6	13.1
swh	luo	2.8	15.0	12.2
eng	som	8.7	20.8	12.1
Average (72 pairs)		15.1	22.6	7.5

Table 7: The top-10 language pairs with the largest improvement over last year's best result. Average refers to all 72 language pairs.

On the other hand, 5 language pairs do not improve from last year. Importantly, though, the largest drop is a mere 1.9 BLEU points for English to Afrikaans, and around 1 BLEU point reduction for the opposite direction as well as English to Kamba. For the few cases where we indeed observe a reduction, the average reduction is only 1 BLEU point (min: -0.46, max: -1.9).

Source	Target	2021	2022	Δ
eng	afr	47.9	46.0	-1.9
eng	kam	6.9	5.9	-1.0
afr	eng	61.0	60.1	-1.0
fra	lin	20.7	20.0	-0.7
eng	umb	4.6	4.1	-0.5

Table 8: The bottom-5 language pairs with the largest performance degradation over last year's best result.

5.5 X-eng and eng-X results

Rk	System	Human	BLEU	spBLEU	chrF2
Systems handling all language pairs					
Constrained					
1	Tencent$_p$	0.40	26.0	28.5	47.6
	Tencent$_c$		25.8	28.3	47.5
	GMU$_c$		25.9	28.0	46.6
2	GMU$_p$	0.22	25.8	28.0	47.0
3	ANVITA$_p$	0.24	24.3	26.3	44.9
	ANVITA$_c$		23.8	25.9	44.5
4	DENTRA$_p$	0.13	23.2	24.9	43.9
	SPRH-DA		2.5	3.2	14.0
5	SPRH-DA	-1.43	2.5	3.0	13.7
Unconstrained					
U	ByteDance$_p$	0.51	31.2	33.8	52.0
U	ByteDance$_c$		31.2	33.7	52.0
Systems handling partial language pairs					
P	Capetown$_p$	-0.04	25.3	27.4	47.8
P	Masakhane$_p$	-0.01	22.3	24.3	44.3
P	Masakhane$_c$		15.3	17.0	37.5

Table 9: Average results in all X-eng pairs, sorted by spBLEU. We only ranked constrained systems that handle all language pairs. We denote *unconstrained* submissions with U, and systems covering partial language pairs with P.

English-centric results (focusing on translating to and from English) are presented in Tables 10 and 9. The ranking of the system does not change depending on the direction. Note, however, that according to all metrics (including human evaluation) translating out of English and into the African languages is harder than the reverse direction for all systems.

5.6 Results on translation between African languages

Last, we present summary results on the average quality for translating between African languages,

[14] last year: spm tokenizer from FLORES-101; this year: spm tokenizer from FLORES-200.

Rk	System	Human	BLEU	spBLEU	chrF2
Systems handling all language pairs					
Constrained					
1	Tencent$_p$	0.33	14.1	18.8	39.5
	Tencent$_c$		13.9	18.5	39.1
2	DENTRA$_p$	0.22	12.8	16.7	37.5
3	GMU$_p$	0.02	12.0	15.2	35.3
	GMU$_c$		12.0	15.1	35.3
	SPRH-DAI$_c$		0.8	0.9	9.0
4	SPRH-DAI$_p$	-1.38	0.6	0.6	7.1
Unonstrained					
U	ByteDance$_p$	0.55	16.6	22.7	43.5
U	ByteDance$_c$		16.6	22.6	43.4
Systems handling partial language pairs					
P	Capetown$_p$	0.13	16.6	22.1	45.4
P	Masakhane$_p$	0.11	14.5	17.5	39.0
P	Masakhane$_c$		10.2	13.2	34.8

Table 10: Average results in all eng-X pairs, sorted by spBLEU. We only ranked constrained systems that handle all language pairs. We denote *unconstrained* submissions with U, and systems covering partial language pairs with P.

Rk	System	Human	BLEU	spBLEU	chrF2
Systems handling all language pairs					
Constrained					
1	Tencent$_p$	0.40	8.0	11.5	31.0
	Tencent$_c$		8.0	11.4	30.9
2	GMU$_p$	0.33	7.7	10.9	29.9
	GMU$_c$		7.7	10.8	29.9
3	DENTRA	-0.64	2.6	4.2	19.5
Unconstrained					
U	ByteDance$_p$	0.51	10.6	15.5	35.7
U	ByteDance$_c$		10.5	15.4	35.6
Systems handling partial language pairs					
P	Capetown$_p$	0.24	10.2	14.4	37.9

Table 11: Average results in all African-African pairs, sorted by spBLEU. We only ranked constrained systems that handle all language pairs. We denote *unconstrained* submissions with U, and systems covering partial language pairs with P.

shown in Table 11. It is worth noting that, although the automatic metrics score imply that the systems are much worse than in the English-centric directions (compare, for example, spBLEU scores of 22.7 on eng-X to 15.5 on african-to-arfican for the ByteDance system), the human evaluation scores are not too different. For instance, the ByteDance system receives an average human Z-score of 0.55 on eng-X and 0.51 on african-african; the Tencent system, with respective scores of 0.33 and 0.40, receives even higher Z-scores for african-to-african languages.

6 Conclusion and Future Work

In this paper, we presented the results of the second shared task on Large-Scale Machine Translation Evaluation. In this edition of the shared task, we evaluate the progress on massively multilingual translation for African Languages in a non-English-centric way. From our findings, we observe that data is still an important factor in translation performance, and that systems that used more data, either in an unconstrained way, or through data augmentation techniques made the most progress.The quality of the data is also important, as most participants developed a set of heuristics to clean it. We also observed the popularity of pre-trained translation models such as: DeltaLM, M2M-100 and mT5, as most systems used a version of these when developing their final model. We observe that there has been a large progress in the quality of translation in since the last iteration of this shared task: there is an improvement of about 7.5 BLEU points across 72 language pairs, and the average BLEU scores went from 15.09 to 22.60. We observe that it is usually harder to translate into than out of African Languages, and it is particularly difficult to translate into West African Languages like Wolof. We also observed that there has been significant progress translating into and out of Northern Sotho, Hausa, and Somali.

Acknowledgements

We thank Kevin Heffernan for developing LASER3 encoders, Onur Çelebi for work on mined bitext and Kenneth Heafield for work on mined bitext and open sourcing. We are also grateful to Toloka for running the human evaluation, as well as to the annotators who contributed with their expertise. We thank Dmitry Ustalov and Mariya Shmatova for sharing their expertise in human evaluation of machine translation. We thank Microsoft Research for contributing Azure compute credits for some participating teams. We are thankful to GMU's ORC computing cluster, where the automatic system test set generation and evaluation was hosted and ran. Mahfuz Alam, Fahim Faisal, and Antonios Anastasopoulos are generously supported by NSF awards FAI-2040926, IIS-2125466, and a Sponsored Research Award from Meta.

References

Idris Abdulmumin, Michael Coenraad Beukman, Jesujoba Alabi, Chris Chinenye Emezue, Everlyn Asiko Chimoto, Tosin Adewumi, Shamsuddeen Hassan Muhammad, Mofetoluwa Oluwaseun Adeyemi, Oreen Yousuf, Sahib Singh, and Tajuddeen Gwadabe. 2022. Separating grains from the chaff: Using data filtering to improve multilingual translation for low-resourced african languages. In *Proceedings of the Seventh Conference on Machine Translation: Shared Task Papers*.

David Adelani, Jesujoba Alabi, Angela Fan, Julia Kreutzer, Xiaoyu Shen, Machel Reid, Dana Ruiter, Dietrich Klakow, Peter Nabende, Ernie Chang, Tajuddeen Gwadabe, Freshia Sackey, Bonaventure F. P. Dossou, Chris Emezue, Colin Leong, Michael Beukman, Shamsuddeen Muhammad, Guyo Jarso, Oreen Yousuf, Andre Niyongabo Rubungo, Gilles Hacheme, Eric Peter Wairagala, Muhammad Umair Nasir, Benjamin Ajibade, Tunde Ajayi, Yvonne Gitau, Jade Abbott, Mohamed Ahmed, Millicent Ochieng, Anuoluwapo Aremu, Perez Ogayo, Jonathan Mukiibi, Fatoumata Ouoba Kabore, Godson Kalipe, Derguene Mbaye, Allahsera Auguste Tapo, Victoire Memdjokam Koagne, Edwin Munkoh-Buabeng, Valencia Wagner, Idris Abdulmumin, Ayodele Awokoya, Happy Buzaaba, Blessing Sibanda, Andiswa Bukula, and Sam Manthalu. 2022. A few thousand translations go a long way! leveraging pre-trained models for African news translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3053–3070, Seattle, United States. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

Damian Blasi, Antonios Anastasopoulos, and Graham Neubig. 2022. Systematic inequalities in language technology performance across the world's languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5486–5505, Dublin, Ireland. Association for Computational Linguistics.

Hiroki Chida, Yohei Murakami, and Mondheera Pituxcoosuvarn. 2022. Quality control for crowdsourced bilingual dictionary in low-resource languages. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6590–6596, Marseille, France. European Language Resources Association.

Jan Christian Blaise Cruz and Lintang Sutawika. 2022. Samsung research philippines - datasaur ai's submission for the wmt22 large scale multilingual translation task. In *Proceedings of the Seventh Conference on Machine Translation: Shared Task Papers*.

Khalid N. Elmadani, Francois Meyer, and Jan Buys. 2022. University of cape town's WMT22 system: Multilingual machine translation for southern african languages. In *Proceedings of the Seventh Conference on Machine Translation: Shared Task Papers*.

Fahim Faisal and Antonios Anastasopoulos. 2022. Phylogeny-inspired adaptation of multilingual models to new languages. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, online. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. Beyond english-centric multilingual machine translation. *JMLR*.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Kevin Heffernan, Onur Çelebi, and Holger Schwenk. 2022. Bitext mining using distilled sentence representations for low-resource languages. In *EMNLP*.

Md Mahfuz Ibn Alam and Antonios Anastasopoulos. 2022. Language adapters for large-scale mt: The GMU system for the wmt 2022 large-scale machine translation evaluation for african languages shared task. In *Proceedings of the Seventh Conference on Machine Translation: Shared Task Papers*.

Wenxiang Jiao, Zhaopeng Tu, Jiarui Li, Wenxuan Wang, Jen tse Huang, and Shuming Shi. 2022. Tencent's multilingual machine translation system for wmt22 large-scale african languages. In *Proceedings of the Seventh Conference on Machine Translation: Shared Task Papers*.

Ratnesh Joshi, Arindam Chatterjee, and Asif Ekbal. 2021. Towards explainable dialogue system: Explaining intent classification using saliency techniques. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 120–127, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLPAI).

Samta Kamboj, Sunil Kumar Sahu, and Neha Sengupta. 2022. DENTRA: Denoising and translation pre-training for multilingual machine translation. In *Proceedings of the Seventh Conference on Machine Translation: Shared Task Papers*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Vilen N Komissarov. 1990. Theory of translation (linguistic aspects). *Moscow: Vysshaya shkola*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, and Furu Wei. 2021. DeltaLM: encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders. `https://arxiv.org/abs/2106.13736`.

Wilhelmina Nekoto, Vukosi Marivate, Tshinondiwa Matsila, Timi Fasubaa, Taiwo Fagbohungbe, Solomon Oluwole Akinola, Shamsuddeen Muhammad, Salomon Kabongo Kabenamualu, Salomey Osei, Freshia Sackey, Rubungo Andre Niyongabo, Ricky Macharm, Perez Ogayo, Orevaoghene Ahia, Musie Meressa Berhe, Mofetoluwa Adeyemi, Masabata Mokgesi-Selinga, Lawrence Okegbemi, Laura Martinus, Kolawole Tajudeen, Kevin Degila, Kelechi Ogueji, Kathleen Siminyu, Julia Kreutzer, Jason Webster, Jamiil Toure Ali, Jade Abbott, Iroro Orife, Ignatius Ezeani, Idris Abdulkadir Dangana, Herman Kamper, Hady Elsahar, Goodness Duru, Ghollah Kioko, Murhabazi Espoir, Elan van Biljon, Daniel Whitenack, Christopher Onyefuluchi, Chris Chinenye Emezue, Bonaventure F. P. Dossou, Blessing Sibanda, Blessing Bassey, Ayodele Olabiyi, Arshath Ramkilowan, Alp Öktem, Adewale Akinfaderin, and Abdallah Bashir. 2020. Participatory research for low-resourced machine translation: A case study in African languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2144–2160, Online. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint 2207.04672*.

Iroro Orife, Julia Kreutzer, Blessing Sibanda, Daniel Whitenack, Kathleen Siminyu, Laura Martinus, Jamiil Toure Ali, Jade Z. Abbott, Vukosi Marivate, Salomon Kabongo, Musie Meressa, Espoir Murhabazi, Orevaoghene Ahia, Elan Van Biljon, Arshath Ramkilowan, Adewale Akinfaderin, Alp Öktem, Wole Akin, Ghollah Kioko, Kevin Degila, Herman Kamper, Bonaventure Dossou, Chris Emezue, Kelechi Ogueji, and Abdallah Bashir. 2020. Masakhane - machine translation for africa. *CoRR*, abs/2003.11529.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Xian Qian, Kai Hu, Jiaqiang Wang, Yifeng Liu, Xingyuan Pan, Jun Cao, and Mingxuan Wang. 2022. The volctrans system for wmt22 multilingual machine translation task. In *Proceedings of the Seventh Conference on Machine Translation: Shared Task Papers*.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500, Online. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Pavanpankaj Vegi, Sivabhavani J, Biswajit Paul, Prasanna Kumar K R, and Chitra Viswanathan. 2022a. ANVITA multilingual neural machine translation system for african languages. In *Proceedings of the Seventh Conference on Machine Translation: Shared Task Papers*.

Pavanpankaj Vegi, Sivabhavani J, Biswajit Paul, Abhinav Mishra, Prashant Banjare, Prasanna Kumar K R, and Chitra Viswanathan. 2022b. Webcrawl african: A multilingual parallel corpora for african languages. In *Proceedings of the Seventh Conference on Machine Translation: Shared Task Papers*.

Barack Wamkaya Wanjawa, Lilian D. A. Wanzare, Florence Indede, Owen McOnyango, Edward Ombui, and Lawrence Muchemi. 2022. Kencorpus: A kenyan language corpus of swahili, dholuo and luhya for natural language processing tasks. *ArXiv*, abs/2208.12081.

Guillaume Wenzek, Vishrav Chaudhary, Angela Fan, Sahir Gomez, Naman Goyal, Somya Jain, Douwe Kiela, Tristan Thrush, and Francisco Guzmán. 2021. Findings of the WMT 2021 shared task on large-scale multilingual machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 89–99, Online. Association for Computational Linguistics.

Jian Yang, Shuming Ma, Haoyang Huang, Dongdong Zhang, Li Dong, Shaohan Huang, Alexandre Muzio, Saksham Singhal, Hany Hassan, Xia Song, and Furu Wei. 2021. Multilingual machine translation systems from microsoft for wmt21 shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 446–455, Online. Association for Computational Linguistics.

A Appendix: Improvement over Last Year

Table 12 compares the performance of last year's best system to this year's best system for the 72 language pairs that intersect between the two.

Lang Pair	**2021**	**2022**	Δ	**Lang Pair**	**2021**	**2022**	Δ
eng-afr	47.9	46.0	-1.9	fra-wol	5.8	8.3	2.5
eng-amh	23.4	31.1	7.7	lin-fra	19.7	26.6	6.9
eng-hau	16.4	31.5	15.1	swh-fra	28.0	39.9	11.8
eng-ibo	17.9	23.5	5.6	wol-fra	11.2	21.6	10.4
eng-kam	6.9	5.9	-1.0	xho-zul	17.5	21.0	3.5
eng-lug	10.6	15.4	4.8	zul-sna	15.5	17.1	1.6
eng-luo	3.5	16.6	13.1	sna-afr	18.6	21.6	3.0
eng-nso	9.5	30.5	21.1	nso-xho	6.6	17.9	11.3
eng-nya	17.6	20.3	2.7	swh-amh	17.5	24.4	6.9
eng-orm	9.2	14.6	5.4	amh-swh	19.2	26.6	7.4
eng-sna	20.4	21.3	0.9	luo-orm	5.8	9.3	3.5
eng-som	8.7	20.8	12.1	som-amh	11.0	18.9	7.9
eng-swh	32.8	42.0	9.2	orm-som	3.3	12.9	9.6
eng-umb	4.6	4.1	-0.5	swh-luo	2.8	15.0	12.2
eng-xho	20.8	23.0	2.2	amh-luo	2.0	12.0	10.0
eng-yor	3.9	12.3	8.4	luo-som	5.0	12.7	7.7
eng-zul	22.3	28.4	6.1	hau-ibo	10.6	17.9	7.3
afr-eng	61.0	60.1	-1.0	ibo-yor	5.8	9.9	4.1
amh-eng	30.8	39.5	8.7	ibo-hau	10.3	21.6	11.3
hau-eng	22.2	40.0	17.8	yor-ibo	5.7	13.5	7.8
ibo-eng	25.3	36.4	11.1	wol-hau	6.6	14.6	8.0
kam-eng	11.2	18.3	7.1	hau-wol	4.6	8.6	4.0
lug-eng	16.6	26.2	9.6	lug-lin	13.3	14.8	1.5
luo-eng	20.0	27.5	7.5	swh-lug	8.5	13.0	4.5
nso-eng	10.0	43.5	33.5	lin-nya	11.5	13.8	2.3
nya-eng	23.0	32.7	9.7	nya-swh	16.2	23.0	6.8
orm-eng	10.6	28.3	17.7	amh-zul	14.0	18.7	4.7
sna-eng	25.5	31.8	6.4	yor-swh	5.8	17.9	12.1
som-eng	20.8	35.0	14.2	swh-yor	6.2	11.0	4.7
swh-eng	37.8	49.0	11.2	zul-amh	14.7	21.2	6.4
umb-eng	9.4	11.9	2.5	nya-som	5.3	13.9	8.6
xho-eng	30.7	38.3	7.6	som-nya	10.7	14.4	3.7
yor-eng	7.1	25.1	18.0	xho-lug	8.6	12.1	3.5
zul-eng	31.2	41.4	10.2	lug-xho	9.1	13.0	3.9
fra-lin	20.7	20.0	-0.7	wol-swh	8.5	16.5	8.0
fra-swh	24.7	30.8	6.1	swh-wol	5.6	8.5	2.9
				Average (72 pairs)	**15.09**	**22.60**	**7.51**

Table 12: Results on African languages on the FLORES-200 test set from last year to this year.

B Appendix: Comparisons for Masakhane and Capetown Models

Two submitted systems (Masakhane and Capetown) only handled some of our 100 evaluation language pairs. For a fair comparison, Tables 13 and 14 present average scores for the language pairs these systems handle. Overall, systems rankings do not change.

Rank	System	Human	BLEU	spBLEU	chrF2
Systems handling all language pairs					
U	ByteDance_pr*	0.54	25.1	28.9	48.2
U	ByteDance_contr*		25.1	28.8	48.1
1	Tencent_pr	0.37	21.3	24.5	44.6
	Tencent_contr		21.0	24.2	44.2
2	DENTRA_pr	0.23	19.1	21.8	42.2
	GMU_contr		19.5	21.5	41.0
3	GMU_pr	0.07	19.4	21.4	40.8
4	Masakhane_pr	0.05	16.6	19.0	39.3
	Masakhane_contr		11.9	14.1	35.0
Systems handling partial language pairs					
P	ANVITA_pr	0.23	26.9	29.0	48.2
P	ANVITA_contr		26.1	28.2	47.6
P	Capetown_pr	0.00	19.3	23.3	45.3
P	SPRH-DAI_contr		1.6	2.0	11.7
P	SPRH-DAI_pr	-1.42	1.4	1.8	10.6

Table 13: Average results in all pairs where Masakhane system participated, sorted by spBLEU. We only ranked constrained systems that handle all language pairs. We denote *unconstrained* submissions with U, and systems covering partial language pairs with P.

Rank	System	Human	BLEU	spBLEU	chrF2
Systems handling all language pairs					
U	ByteDance_contr*		24.2	29.5	50.0
U	ByteDance_pr*	0.41	24.1	29.5	50.1
1	Tencent_pr	0.41	21.9	27.0	48.6
	Tencent_contr		21.7	26.8	48.4
2	GMU_pr	0.21	20.7	24.8	46.0
	GMU_contr		20.7	24.7	46.0
3	Capetown_pr	0.09	17.4	21.3	43.7
4	DENTRA_pr	0.12	17.5	21.1	41.5
Systems handling partial language pairs					
P	ANVITA_pr	0.36	32.0	34.3	52.7
P	ANVITA_contr		31.6	33.9	52.4
P	Masakhane_pr	0.05	19.2	23.1	44.9
P	Masakhane_contr		11.1	14.0	35.8
P	SPRH-DAI_pr	-1.45	2.2	2.6	12.8
P	SPRH-DAI_contr		2.1	2.5	13.0

Table 14: Average results in all pairs where CapeTown system participated, sorted by spBLEU. We only ranked constrained systems that handle all language pairs. We denote *unconstrained* submissions with U, and systems covering partial language pairs with P.

C Appendix: Human Evaluation Guidelines

C.1 About

In the interface you will see a source sentence and two translations. Your task is to evaluate the quality of translations on a 0-100 scale where 0 is ridiculously bad and 100 is a perfect translation.

By doing this you will help us a lot to improve the quality of machine translation for African languages in all their glory and diversity.

C.2 How To Evaluate

The evaluation slider can go from 0 to 100. While choosing the most appropriate score, please consider the following features of a good translation starting from the most important:

1. Acceptable translation is grammatically correct, looks natural and makes sense to the reader.
2. Acceptable translation also has the same general meaning and communicative goal (what did it want to say?) as the source sentence
3. OK translation must be completely fluent and natural in addition to the above
4. Good translation keeps the style of the source sentence (e.g. formal or informal, colloquial style) in addition to being fluent and conveying the same meaning
5. Great translation keeps as much meaning and nuances as the source sentence in addition to being perfectly fluent.
6. Amazing translation also chooses the same means to describe the situation as the source sentence wherever the destination language has the same ways of doing it: the same metaphors, set expressions and such.

While performing the task, please do not use any automatic translation as the goal is to evaluate it with the help of human experts. You can use a dictionary if you found a word that you don't know.

C.2.1 Tips

- Evaluate the phrases as a whole and not word by word
- Check if any meaning was lost or unnecessarily added
- Check if there are any grammatical mistakes
- Check if anything remained untranslated
- Check if set expressions and metaphors were translated simply word by word (bad) or as a whole and according to the expressions used in the destination language (good)
- Use your own feeling as the speaker: do you consider the translation good, natural, clear and easily understandable

Language Pairs used in Evaluation			
eng-afr	eng-xho	som-eng	swh-fra
eng-amh	eng-yor	ssw-eng	xho-zul
eng-hau	eng-zul	swh-eng	zul-sna
eng-ibo	afr-eng	tsn-eng	afr-ssw
eng-lug	amh-eng	tso-eng	ssw-tsn
eng-nya	hau-eng	xho-eng	tsn-tso
eng-orm	ibo-eng	yor-eng	hau-ibo
eng-kin	lug-eng	zul-eng	ibo-yor
eng-sna	nya-eng	fra-lin	ibo-hau
eng-ssw	orm-eng	fra-swh	yor-ibo
eng-swh	kin-eng	fra-wol	swh-lug
eng-tsn	sna-eng	lin-fra	

Table 15: List of languages used for human evalulation

D Appendix: Individual Language Pair Results

System	Rank	spBLEU	BLEU	chrF2
Tencent_pr	1-3	46.0	40.2	68.0
ByteDance_pr	1-3	45.9	39.6	67.9
ByteDance_contr	1-3	45.7	39.3	67.8
Tencent_contr	4-6	45.5	40.0	67.6
DENTRA_pr	4-6	45.2	39.7	67.8
GMU_contr	4-6	45.2	39.8	67.5
GMU_pr	7	44.9	39.6	67.3
CapeTown_pr	8	40.4	35.9	64.3
SPRH-DAI_pr	9	4.3	4.0	21.8
SPRH-DAI_contr	10	3.0	2.6	19.6

Table 16: Results in eng-afr, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	31.1	12.9	42.4
ByteDance_pr	1-2	30.9	12.5	42.3
Tencent_pr	3	23.5	8.2	37.6
GMU_contr	4-5	22.0	7.6	35.5
Tencent_contr	4-5	21.9	7.7	36.5
GMU_pr	6	21.4	7.3	35.0
DENTRA_pr	7	15.2	5.2	29.4
SPRH-DAI_contr	8	0.2	0.3	0.5
SPRH-DAI_pr	9	0.0	0.1	2.9

Table 17: Results in eng-amh, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-2	5.4	3.8	24.7
ByteDance_contr	1-2	5.3	3.8	24.5
SPRH-DAI_contr	3-4	1.2	0.9	11.9
DENTRA_pr	3-4	1.1	0.7	14.0
Tencent_pr	5-6	0.9	0.5	14.8
Tencent_contr	5-6	0.8	0.4	14.7
GMU_pr	7-8	0.5	0.3	14.4
GMU_contr	7-8	0.5	0.3	14.0
SPRH-DAI_pr	9	0.1	0.1	4.8

Table 18: Results in eng-fuv, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	31.4	28.3	54.9
ByteDance_contr	2	30.9	27.9	54.8
Tencent_contr	3-4	28.7	25.4	53.9
Tencent_pr	3-4	28.7	25.5	53.9
DENTRA_pr	5	25.0	22.9	51.4
Masakhane_pr	6	19.7	17.7	45.8
Masakhane_contr	7	12.2	10.7	38.4
GMU_pr	8	5.2	14.5	30.0
GMU_contr	9	4.5	13.3	27.7
SPRH-DAI_contr	10	1.0	0.5	10.8
SPRH-DAI_pr	11	0.5	0.3	9.7

Table 19: Results in eng-hau, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
Tencent_pr	1	23.6	20.1	45.9
ByteDance_pr	2-4	23.2	19.7	45.6
ByteDance_contr	2-4	23.0	19.6	45.5
Tencent_contr	2-4	23.0	19.6	45.3
GMU_pr	5-6	19.6	17.3	42.4
GMU_contr	5-6	19.6	17.3	42.7
DENTRA_pr	7	18.0	16.4	42.0
Masakhane_pr	8	17.7	15.2	40.4
Masakhane_contr	9	14.7	11.9	36.5
SPRH-DAI_contr	10	0.8	0.6	10.4
SPRH-DAI_pr	11	0.1	0.2	5.0

Table 20: Results in eng-ibo, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	5.9	4.7	26.2
ByteDance_contr	2	5.3	4.1	25.4
GMU_pr	3-4	4.0	3.0	21.6
GMU_contr	3-4	4.0	2.9	21.6
DENTRA_pr	5-6	3.0	2.5	20.8
Tencent_pr	5-6	2.8	2.0	20.9
Tencent_contr	7	2.5	1.8	20.2
SPRH-DAI_contr	8	0.8	0.6	9.3
SPRH-DAI_pr	9	0.1	0.1	3.0

Table 21: Results in eng-kam, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	15.4	9.8	45.5
ByteDance_contr	2	15.2	9.7	45.4
DENTRA_pr	3-5	7.8	5.2	35.2
Tencent_pr	3-5	7.8	5.9	36.5
GMU_contr	3-5	7.5	5.8	34.8
GMU_pr	6-7	7.0	5.6	33.6
Tencent_contr	6-7	7.0	5.5	35.2
Masakhane_contr	8	6.2	4.5	33.1
Masakhane_pr	9	5.4	4.6	31.0
SPRH-DAI_contr	10	0.9	1.1	10.0
SPRH-DAI_pr	11	0.3	0.2	4.7

Table 22: Results in eng-lug, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	16.6	12.6	42.6
ByteDance_contr	2	16.4	12.4	42.4
GMU_pr	3-4	10.4	8.1	33.8
GMU_contr	3-4	10.4	8.0	33.9
DENTRA_pr	5	7.7	6.1	29.7
Tencent_pr	6	6.5	4.9	28.3
Tencent_contr	7	5.7	4.4	27.2
SPRH-DAI_contr	8	1.4	1.0	11.9
SPRH-DAI_pr	9	0.6	0.5	6.3

Table 23: Results in eng-luo, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	30.5	27.9	54.6
ByteDance_pr	1-2	30.4	27.7	54.6
Tencent_pr	3	28.5	25.4	53.8
Tencent_contr	4	28.0	24.9	53.3
DENTRA_pr	5	26.3	24.7	51.5
GMU_contr	6	24.8	23.5	49.9
GMU_pr	7-8	24.4	23.0	49.1
CapeTown_pr	7-8	24.1	22.7	50.0
SPRH-DAI_contr	9	0.9	0.4	9.7
SPRH-DAI_pr	10	0.4	0.3	6.0

Table 24: Results in eng-nso, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-3	20.3	15.4	50.2
ByteDance_contr	1-3	20.2	15.4	50.2
Tencent_contr	1-3	20.0	15.6	51.4
Tencent_pr	4	19.8	15.3	51.3
GMU_pr	5-6	17.2	13.4	48.4
GMU_contr	5-6	17.2	13.3	48.5
DENTRA_pr	7	16.3	13.3	48.0
SPRH-DAI_contr	8	1.4	1.4	13.2
SPRH-DAI_pr	9	0.8	1.1	12.2

Table 25: Results in eng-nya, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	14.6	6.7	45.1
ByteDance_pr	1-2	14.6	6.8	45.0
DENTRA_pr	3	4.4	2.1	28.3
Tencent_pr	4	2.8	1.4	25.2
GMU_pr	5-7	2.4	1.5	21.4
GMU_contr	5-7	2.4	1.4	21.6
Tencent_contr	5-7	2.4	1.2	23.4
SPRH-DAI_contr	8	0.1	0.1	5.8
SPRH-DAI_pr	9	0.0	0.0	2.9

Table 26: Results in eng-orm, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	29.6	23.3	55.5
ByteDance_contr	2	29.4	23.2	55.2
Tencent_pr	3	22.6	18.1	49.4
Tencent_contr	4	21.9	17.8	48.7
DENTRA_pr	5	18.5	14.4	45.8
GMU_contr	6	16.4	13.2	41.7
GMU_pr	7	16.2	12.8	41.2
SPRH-DAI_contr	8	0.4	0.4	9.6
SPRH-DAI_pr	9	0.4	0.3	6.4

Table 27: Results in eng-kin, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1	21.3	13.3	47.8
ByteDance_pr	2-3	21.1	13.1	47.8
Tencent_pr	2-3	20.8	12.9	49.3
Tencent_contr	4	20.5	12.7	49.1
DENTRA_pr	5	18.8	11.9	47.4
CapeTown_pr	6	17.6	10.3	46.4
GMU_contr	7-8	16.8	10.6	46.1
GMU_pr	7-8	16.7	10.6	46.1
SPRH-DAI_contr	9	0.8	1.0	10.8
SPRH-DAI_pr	10	0.8	1.1	13.9

Table 28: Results in eng-sna, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	20.8	15.0	48.2
ByteDance_pr	1-2	20.8	14.9	48.2
Tencent_pr	3	17.8	12.2	47.1
GMU_contr	4-6	17.5	11.9	45.8
Tencent_contr	4-6	17.5	11.9	46.9
GMU_pr	4-6	17.3	11.9	45.7
DENTRA_pr	7	15.1	10.4	43.5
SPRH-DAI_contr	8	0.5	0.4	8.8
SPRH-DAI_pr	9	0.1	0.3	8.0

Table 29: Results in eng-som, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	22.2	11.1	52.2
ByteDance_pr	1-2	22.0	11.3	52.6
Tencent_pr	3	19.2	9.6	49.4
Tencent_contr	4	18.8	9.5	48.9
DENTRA_pr	5	16.4	8.3	46.0
CapeTown_pr	6	15.5	7.6	44.9
GMU_contr	7	15.1	7.2	45.4
GMU_pr	8	14.4	6.8	44.4
SPRH-DAI_contr	9-10	0.8	1.1	10.8
SPRH-DAI_pr	9-10	0.8	0.7	10.9

Table 30: Results in eng-ssw, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	42.0	37.4	63.8
ByteDance_pr	1-2	42.0	37.2	63.7
Tencent_pr	3	39.2	34.8	62.8
Tencent_contr	4	38.8	34.4	62.5
DENTRA_pr	5	37.2	33.3	61.6
GMU_contr	6	36.5	32.7	61.2
GMU_pr	7	36.2	32.6	60.8
Masakhane_pr	8	35.1	31.5	60.2
Masakhane_contr	9	27.8	24.3	55.0
SPRH-DAI_contr	10	1.6	1.1	15.2
SPRH-DAI_pr	11	1.4	1.3	17.0

Table 31: Results in eng-swh, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	28.5	25.9	53.2
ByteDance_pr	1-2	28.4	25.9	53.1
Tencent_pr	3-4	25.6	22.9	50.3
Tencent_contr	3-4	25.3	22.7	49.9
DENTRA_pr	5	21.2	20.1	46.7
GMU_contr	6	20.7	19.7	46.0
GMU_pr	7	20.1	19.1	45.3
CapeTown_pr	8	19.7	18.8	45.3
Masakhane_pr	9	19.0	17.8	44.2
Masakhane_contr	10	11.0	10.1	36.0
SPRH-DAI_contr	11	0.7	0.3	9.1
SPRH-DAI_pr	12	0.3	0.3	5.9

Table 32: Results in eng-tsn, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	28.0	23.3	53.1
ByteDance_pr	1-2	27.8	23.0	53.2
Tencent_contr	3-4	21.7	18.8	49.5
Tencent_pr	3-4	21.4	18.8	49.6
GMU_contr	5-6	20.5	17.4	47.6
GMU_pr	5-6	20.1	17.2	47.2
DENTRA_pr	7	19.2	16.9	45.8
CapeTown_pr	8	17.9	15.8	44.7
SPRH-DAI_contr	9	1.1	0.7	10.5
SPRH-DAI_pr	10	0.4	0.3	4.2

Table 33: Results in eng-tso, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	4.1	2.0	31.3
ByteDance_contr	2	4.0	2.0	30.3
GMU_pr	3-4	2.2	0.9	23.3
GMU_contr	3-4	2.1	0.8	22.8
DENTRA_pr	5	2.0	1.2	22.9
Tencent_pr	6	1.8	0.9	22.6
Tencent_contr	7	1.6	0.9	21.8
SPRH-DAI_contr	8	0.7	0.6	9.8
SPRH-DAI_pr	9	0.2	0.3	3.7

Table 34: Results in eng-umb, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	23.0	12.1	51.2
ByteDance_pr	1-2	22.9	12.1	51.0
Tencent_contr	3	22.3	11.7	52.1
Tencent_pr	4	21.9	11.4	52.0
DENTRA_pr	5	20.2	10.2	50.1
CapeTown_pr	6	18.6	9.4	48.7
GMU_pr	7	3.5	1.4	20.2
GMU_contr	8	2.7	1.0	17.5
SPRH-DAI_pr	9	0.9	0.6	13.5
SPRH-DAI_contr	10	0.6	0.6	13.4

Table 35: Results in eng-xho, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	12.3	5.6	28.2
ByteDance_pr	1-2	12.3	5.5	28.2
Masakhane_contr	3	7.7	4.2	23.5
Tencent_pr	4	6.5	3.4	22.7
Tencent_contr	5	5.7	3.0	21.8
Masakhane_pr	6-8	5.0	3.2	21.5
GMU_pr	6-8	4.9	3.2	21.9
GMU_contr	6-8	4.8	3.2	21.8
DENTRA_pr	9	4.4	3.1	21.8
SPRH-DAI_contr	10	0.3	0.3	7.5
SPRH-DAI_pr	11	0.0	0.1	5.1

Table 36: Results in eng-yor, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-2	28.4	16.4	54.9
ByteDance_contr	1-2	28.3	16.4	54.9
Tencent_pr	3	26.8	14.9	55.3
Tencent_contr	4	26.4	14.5	55.0
GMU_pr	5-7	24.0	13.5	53.5
DENTRA_pr	5-7	24.0	12.7	53.1
GMU_contr	5-7	23.9	13.2	53.6
CapeTown_pr	8	22.8	11.9	52.0
Masakhane_pr	9	20.9	11.0	50.6
Masakhane_contr	10	13.1	6.0	41.8
SPRH-DAI_pr	11	0.8	0.5	13.0
SPRH-DAI_contr	12	0.5	0.6	12.4

Table 37: Results in eng-zul, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-4	60.1	57.4	75.5
ByteDance_pr	1-4	60.0	57.4	75.6
GMU_contr	1-4	60.0	56.9	76.1
GMU_pr	1-4	59.9	57.0	76.1
ANVITA_contr	5	58.7	55.7	75.5
Tencent_pr	6	58.1	55.0	75.2
DENTRA_pr	7-8	57.4	54.6	74.5
Tencent_contr	7-8	57.4	54.3	74.8
CapeTown_pr	9	46.4	44.6	67.4
SPRH-DAI_pr	10	9.0	8.3	27.3
SPRH-DAI_contr	11	7.1	6.1	22.9

Table 38: Results in afr-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	39.5	36.3	61.6
ByteDance_contr	2	39.2	36.2	61.4
GMU_contr	3	32.2	30.7	55.7
GMU_pr	4	31.6	30.1	55.3
Tencent_contr	5	29.0	27.6	53.6
Tencent_pr	6	28.6	26.7	53.2
ANVITA_contr	7	25.2	24.1	49.4
DENTRA_pr	8	23.3	22.5	48.4
SPRH-DAI_contr	9	1.0	0.9	12.4
SPRH-DAI_pr	10	0.8	0.7	12.2

Table 39: Results in amh-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-2	13.8	11.4	33.4
ByteDance_contr	1-2	13.7	11.3	33.2
GMU_pr	3-6	8.5	6.7	24.3
GMU_contr	3-6	8.5	6.9	25.0
Tencent_pr	3-6	8.5	6.5	25.2
DENTRA_pr	3-6	8.2	6.6	24.2
Tencent_contr	7-8	8.1	6.2	25.0
ANVITA_contr	7-8	7.9	6.1	23.4
SPRH-DAI_contr	9	2.0	1.4	12.1
SPRH-DAI_pr	10	1.8	1.3	11.7

Table 40: Results in fuv-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-2	40.0	37.7	58.8
ByteDance_contr	1-2	39.8	37.6	58.7
Tencent_pr	3	33.5	30.6	54.9
Tencent_contr	4-5	33.2	30.3	54.7
GMU_contr	4-5	32.4	29.6	52.6
GMU_pr	6-7	31.8	29.1	52.1
ANVITA_contr	6-7	31.3	28.8	52.3
DENTRA_pr	8	30.4	28.2	51.5
Masakhane_pr	9	25.1	22.7	46.5
Masakhane_contr	10	17.3	15.6	39.4
SPRH-DAI_pr	11-12	3.7	2.7	15.8
SPRH-DAI_contr	11-12	3.6	2.5	15.7

Table 41: Results in hau-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	36.4	33.6	56.7
ByteDance_pr	1-2	36.4	33.5	56.8
GMU_pr	3-4	30.8	28.2	51.6
GMU_contr	3-4	30.8	28.3	51.8
Tencent_contr	5	29.1	26.8	51.7
Tencent_pr	6	28.6	26.3	51.2
ANVITA_contr	7	25.8	23.6	47.5
DENTRA_pr	8	25.2	22.6	47.1
Masakhane_pr	9	23.2	20.9	45.9
Masakhane_contr	10	16.8	15.0	39.6
SPRH-DAI_contr	11	3.0	2.1	13.7
SPRH-DAI_pr	12	2.6	1.9	13.2

Table 42: Results in ibo-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	18.3	15.7	35.9
ByteDance_pr	1-2	18.3	15.7	36.6
GMU_pr	3-4	13.2	10.8	30.6
GMU_contr	3-4	13.2	10.9	30.5
ANVITA_contr	5-6	12.4	10.3	29.9
Tencent_pr	5-6	12.3	9.9	30.5
DENTRA_pr	7-8	11.9	9.8	29.2
Tencent_contr	7-8	11.7	9.4	30.3
SPRH-DAI_contr	9-10	2.9	2.1	13.6
SPRH-DAI_pr	9-10	2.8	2.1	13.2

Table 43: Results in kam-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-2	26.2	24.2	46.8
ByteDance_contr	1-2	26.1	24.2	46.6
Tencent_pr	3-4	21.8	19.7	41.9
Tencent_contr	3-4	21.6	19.7	41.8
GMU_pr	5	19.0	17.2	38.3
GMU_contr	6-7	18.7	16.8	37.8
ANVITA_contr	6-7	18.5	16.5	38.0
DENTRA_pr	8	18.1	16.4	37.7
Masakhane_pr	9	16.9	14.9	36.8
Masakhane_contr	10	13.9	12.2	35.0
SPRH-DAI_contr	11	2.6	2.0	14.1
SPRH-DAI_pr	12	2.4	1.8	13.0

Table 44: Results in lug-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	27.5	24.8	48.6
ByteDance_pr	1-2	27.4	24.7	48.6
Tencent_pr	3	22.0	19.3	43.0
Tencent_contr	4-6	21.5	18.8	42.8
GMU_contr	4-6	21.2	19.2	41.0
GMU_pr	4-6	21.0	19.1	40.7
DENTRA_pr	7-8	19.6	17.9	39.2
ANVITA_contr	7-8	19.5	17.6	39.3
SPRH-DAI_contr	9	2.4	1.8	12.7
SPRH-DAI_pr	10	2.2	1.8	12.0

Table 45: Results in luo-eng, sorted by spBLEU.

System	**Rank**	**spBLEU**	**BLEU**	**chrF2**
ByteDance_pr	1	43.5	41.2	61.0
ByteDance_contr	2	43.1	40.7	60.7
Tencent_pr	3	39.6	37.1	58.5
Tencent_contr	4	39.2	36.7	58.2
GMU_pr	5	37.1	35.2	56.2
GMU_contr	6	36.7	34.6	55.8
ANVITA_contr	7	35.5	33.7	54.5
DENTRA_pr	8	33.8	32.2	53.8
CapeTown_pr	9	28.0	26.5	49.3
SPRH-DAI_contr	10	4.1	3.1	16.1
SPRH-DAI_pr	11	3.6	2.8	15.3

Table 46: Results in nso-eng, sorted by spBLEU.

System	**Rank**	**spBLEU**	**BLEU**	**chrF2**
ByteDance_contr	1-2	32.7	28.9	53.0
ByteDance_pr	1-2	32.6	28.9	52.9
GMU_pr	3-4	29.0	25.8	49.6
GMU_contr	3-4	28.8	25.8	49.5
Tencent_pr	5-6	28.0	24.6	48.9
Tencent_contr	5-6	27.9	24.4	49.1
ANVITA_contr	7	26.2	23.1	47.1
DENTRA_pr	8	25.3	22.7	46.3
SPRH-DAI_contr	9	4.1	3.1	17.1
SPRH-DAI_pr	10	3.9	3.0	16.5

Table 47: Results in nya-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-2	28.3	26.3	50.9
ByteDance_contr	1-2	28.0	26.0	50.7
Tencent_contr	3-4	17.6	16.6	40.9
Tencent_pr	3-4	17.6	16.6	40.6
GMU_contr	5-6	15.3	14.6	36.8
GMU_pr	5-6	15.2	14.6	36.6
ANVITA_contr	7-8	12.0	11.2	32.9
DENTRA_pr	7-8	11.8	11.3	32.8
SPRH-DAI_contr	9	0.9	0.6	10.2
SPRH-DAI_pr	10	0.7	0.5	9.5

Table 48: Results in orm-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	38.7	35.8	58.5
ByteDance_pr	1-2	38.7	36.0	58.6
Tencent_pr	3-4	32.8	30.6	53.7
Tencent_contr	3-4	32.6	30.5	53.6
GMU_contr	5	30.2	28.4	51.3
GMU_pr	6	30.0	28.3	51.1
ANVITA_contr	7	28.9	27.4	50.1
DENTRA_pr	8	26.1	24.8	47.2
SPRH-DAI_contr	9	3.4	2.7	16.0
SPRH-DAI_pr	10	3.1	2.3	15.1

Table 49: Results in kin-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1	31.8	28.1	52.0
ByteDance_pr	2	31.4	27.8	51.8
GMU_pr	3-5	29.4	26.3	49.8
GMU_contr	3-5	29.3	26.1	49.5
Tencent_pr	3-5	29.1	25.6	49.6
Tencent_contr	6	28.5	24.9	49.4
ANVITA_contr	7	27.6	24.6	47.7
DENTRA_pr	8	26.0	23.2	46.9
CapeTown_pr	9	22.1	18.7	44.6
SPRH-DAI_contr	10-11	3.7	3.0	16.6
SPRH-DAI_pr	10-11	3.7	3.0	16.1

Table 50: Results in sna-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	35.0	32.9	55.1
ByteDance_contr	2	34.6	32.5	54.7
Tencent_pr	3-4	28.4	26.5	49.8
Tencent_contr	3-4	28.2	26.4	49.8
GMU_pr	5-6	27.8	26.4	48.1
GMU_contr	5-6	27.8	26.4	48.0
DENTRA_pr	7	23.4	21.9	44.6
ANVITA_contr	8	22.2	20.6	42.7
SPRH-DAI_contr	9	3.0	2.3	15.0
SPRH-DAI_pr	10	2.5	2.0	13.6

Table 51: Results in som-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1	36.8	34.3	56.5
ByteDance_pr	2	36.5	33.9	56.2
Tencent_pr	3	32.9	30.1	53.0
Tencent_contr	4	32.3	29.5	52.8
GMU_pr	5-6	29.2	27.1	49.8
GMU_contr	5-6	29.0	27.1	49.5
ANVITA_contr	7-8	27.5	25.5	47.5
DENTRA_pr	7-8	27.5	25.8	48.5
CapeTown_pr	9	23.5	21.5	45.2
SPRH-DAI_pr	10	3.5	2.6	14.9
SPRH-DAI_contr	11	3.3	2.6	15.0

Table 52: Results in ssw-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	49.0	47.4	67.7
ByteDance_contr	2	48.6	47.0	67.4
GMU_pr	3-6	42.8	41.0	62.8
Tencent_pr	3-6	42.8	41.1	62.9
GMU_contr	3-6	42.7	41.1	62.8
Tencent_contr	3-6	42.7	40.9	62.8
ANVITA_contr	7	41.7	40.4	62.2
DENTRA_pr	8	39.8	38.9	60.7
Masakhane_pr	9	36.4	35.2	58.4
Masakhane_contr	10	28.2	27.5	51.7
SPRH-DAI_contr	11-12	4.5	3.9	18.8
SPRH-DAI_pr	11-12	4.4	4.1	19.0

Table 53: Results in swh-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1	34.4	31.8	54.5
ByteDance_pr	2	34.2	31.6	54.4
Tencent_pr	3	30.3	27.5	51.7
Tencent_contr	4	29.8	26.9	51.5
GMU_pr	5	29.3	26.6	50.3
GMU_contr	6	28.2	25.6	48.9
ANVITA_contr	7	27.5	25.4	48.5
DENTRA_pr	8	26.2	23.9	47.6
Masakhane_pr	9	23.5	21.2	45.1
CapeTown_pr	10	22.1	19.8	44.2
Masakhane_contr	11	11.3	9.7	33.3
SPRH-DAI_contr	12	3.3	2.6	15.1
SPRH-DAI_pr	13	2.9	2.3	14.4

Table 54: Results in tsn-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	35.6	32.8	54.6
ByteDance_pr	1-2	35.6	32.8	54.7
Tencent_contr	3-4	31.9	29.6	52.1
Tencent_pr	3-4	31.9	29.7	52.1
GMU_pr	5-6	30.4	28.3	50.6
GMU_contr	5-6	30.2	28.1	50.2
ANVITA_contr	7-8	27.2	25.3	47.3
DENTRA_pr	7-8	27.2	25.6	47.7
CapeTown_pr	9	21.8	20.3	43.2
SPRH-DAI_contr	10	3.0	2.4	14.3
SPRH-DAI_pr	11	2.8	2.1	13.4

Table 55: Results in tso-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	11.9	9.7	31.0
ByteDance_pr	1-2	11.8	9.7	31.3
GMU_pr	3	9.9	8.0	28.0
GMU_contr	4-6	9.6	7.7	27.7
Tencent_contr	4-6	9.4	7.0	27.7
Tencent_pr	4-6	9.4	7.1	27.6
ANVITA_contr	7	8.1	6.2	26.4
DENTRA_pr	8	7.4	5.8	25.1
SPRH-DAI_contr	9-10	1.5	0.9	12.3
SPRH-DAI_pr	9-10	1.5	1.0	11.8

Table 56: Results in umb-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-2	38.3	35.4	57.7
ByteDance_contr	1-2	38.2	35.3	57.6
Tencent_pr	3-5	34.2	31.1	54.6
Tencent_contr	3-5	34.1	31.0	54.4
GMU_contr	3-5	33.9	31.3	54.3
GMU_pr	6	33.7	31.3	54.2
ANVITA_contr	7	32.4	29.8	52.9
DENTRA_pr	8	30.8	28.8	51.8
CapeTown_pr	9	26.7	24.3	49.2
SPRH-DAI_pr	10-11	4.1	3.3	17.5
SPRH-DAI_contr	10-11	4.0	3.2	17.5

Table 57: Results in xho-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-2	25.1	22.9	46.4
ByteDance_contr	1-2	24.9	22.6	46.0
Tencent_contr	3-4	20.2	17.4	41.3
Tencent_pr	3-4	20.2	17.5	41.5
GMU_contr	5-6	19.7	17.6	40.2
GMU_pr	5-6	19.6	17.6	40.1
ANVITA_contr	7	17.9	15.8	38.5
DENTRA_pr	8	16.4	14.5	37.2
Masakhane_pr	9	16.1	14.2	36.9
Masakhane_contr	10	10.9	8.8	31.2
SPRH-DAI_contr	11	2.5	1.8	13.3
SPRH-DAI_pr	12	2.1	1.5	12.1

Table 58: Results in yor-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	41.4	39.2	60.4
ByteDance_pr	1-2	41.3	39.0	60.4
GMU_pr	3-5	36.8	34.6	56.9
GMU_contr	3-5	36.8	34.4	56.9
Tencent_pr	3-5	36.6	34.0	56.6
Tencent_contr	6	36.1	33.3	56.4
ANVITA_contr	7	34.9	32.5	55.2
DENTRA_pr	8	32.7	31.1	53.5
Masakhane_pr	9	29.0	26.8	50.5
CapeTown_pr	10	28.5	26.7	50.7
Masakhane_contr	11	20.4	18.5	43.1
SPRH-DAI_contr	12-13	3.5	2.9	16.5
SPRH-DAI_pr	12-13	3.5	2.9	16.4

Table 59: Results in zul-eng, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	25.3	18.4	52.1
ByteDance_contr	2	25.1	18.3	51.9
Tencent_pr	3-4	19.3	14.1	47.0
Tencent_contr	3-4	19.0	14.4	46.6
DENTRA_pr	5-6	14.9	10.7	41.8
GMU_contr	5-6	14.4	10.3	41.3
GMU_pr	7	13.9	10.1	40.4

Table 60: Results in fra-kin, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	20.0	16.6	50.8
ByteDance_contr	2	19.7	16.2	50.7
DENTRA_pr	3	17.1	14.7	45.9
Tencent_pr	4-5	16.4	14.1	46.1
Tencent_contr	4-5	16.3	13.8	45.8
GMU_pr	6-7	10.1	7.5	37.4
GMU_contr	6-7	10.1	7.2	37.1

Table 61: Results in fra-lin, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	30.7	25.0	56.3
ByteDance_contr	2	30.2	24.5	56.0
Tencent_pr	3	29.2	24.1	55.2
Tencent_contr	4	28.8	23.8	54.9
GMU_pr	5	27.6	23.4	53.2
GMU_contr	6	27.1	22.8	52.8
DENTRA_pr	7	24.9	21.2	50.8

Table 62: Results in fra-swh, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
DENTRA_pr	1	8.3	5.6	28.2
ByteDance_pr	2-3	7.3	5.2	27.6
ByteDance_contr	2-3	7.1	5.0	27.5
Masakhane_contr	4	6.3	4.4	27.4
Tencent_pr	5	4.8	3.9	23.2
Tencent_contr	6	4.2	3.4	22.0
GMU_pr	7	3.0	2.0	15.9
GMU_contr	8	2.6	1.8	14.0
Masakhane_pr	9	2.2	1.5	19.5

Table 63: Results in fra-wol, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	33.7	29.1	54.1
ByteDance_pr	1-2	33.7	29.1	54.2
Tencent_pr	3	27.2	23.6	49.4
Tencent_contr	4	26.8	23.1	49.2
GMU_pr	5	26.4	23.0	48.0
GMU_contr	6	26.0	22.7	47.8
DENTRA_pr	7	22.2	18.5	43.4

Table 64: Results in kin-fra, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	26.6	22.7	46.9
ByteDance_pr	1-2	26.5	22.6	46.8
Tencent_contr	3-4	23.8	19.7	44.8
Tencent_pr	3-4	23.6	19.6	44.6
GMU_contr	5-6	22.9	19.1	43.1
GMU_pr	5-6	22.8	18.8	42.7
DENTRA_pr	7	20.7	16.9	41.7

Table 65: Results in lin-fra, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	39.9	35.5	60.0
ByteDance_contr	2	39.6	35.2	59.7
GMU_contr	3-4	35.0	31.0	56.0
GMU_pr	3-4	34.9	30.8	56.0
Tencent_pr	5-6	33.6	29.2	55.1
Tencent_contr	5-6	33.4	29.1	55.0
DENTRA_pr	7	31.1	26.8	53.0

Table 66: Results in swh-fra, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	21.6	17.8	41.7
ByteDance_contr	2	21.2	17.6	41.2
Tencent_contr	3-5	14.6	12.0	36.0
Tencent_pr	3-5	14.5	12.1	35.8
DENTRA_pr	3-5	14.4	11.2	35.1
GMU_contr	6-7	13.2	10.5	31.1
GMU_pr	6-7	13.0	10.3	30.6
Masakhane_pr	8	9.2	7.5	29.7
Masakhane_contr	9	8.3	7.0	30.2

Table 67: Results in wol-fra, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-3	21.0	10.3	48.1
ByteDance_contr	1-3	20.9	10.4	47.9
Tencent_pr	1-3	20.5	9.9	49.2
Tencent_contr	4-5	20.4	9.8	49.0
GMU_contr	4-5	20.1	10.0	49.1
GMU_pr	6	20.0	9.9	48.9
CapeTown_pr	7	18.0	8.5	47.4
DENTRA_pr	8	10.6	4.1	38.2

Table 68: Results in xho-zul, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-5	17.1	10.1	44.1
ByteDance_pr	1-5	17.1	10.0	44.3
Tencent_contr	1-5	17.0	9.9	45.7
Tencent_pr	1-5	17.0	10.0	45.8
GMU_pr	1-5	16.7	9.9	45.8
GMU_contr	6	16.6	9.9	45.8
CapeTown_pr	7	15.0	8.5	44.2
DENTRA_pr	8	4.4	2.1	25.2

Table 69: Results in zul-sna, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	21.6	17.3	45.6
ByteDance_pr	1-2	21.5	17.2	45.4
GMU_pr	3-4	20.1	17.0	43.8
GMU_contr	3-4	20.0	17.0	43.8
Tencent_contr	5-6	19.1	15.7	43.1
Tencent_pr	5-6	19.0	15.5	43.2
CapeTown_pr	7	15.1	12.0	40.1
DENTRA_pr	8	13.9	11.2	38.1

Table 70: Results in sna-afr, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-2	18.9	9.2	48.7
ByteDance_contr	1-2	18.7	8.9	48.4
Tencent_pr	3	17.3	7.8	47.9
Tencent_contr	4	16.6	7.9	46.5
GMU_contr	5	14.7	6.7	45.2
GMU_pr	6	14.3	6.5	44.7
CapeTown_pr	7	11.2	5.4	40.0
DENTRA_pr	8	8.3	4.3	36.1

Table 71: Results in afr-ssw, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	21.6	19.1	46.8
ByteDance_pr	1-2	21.6	19.2	46.9
Tencent_contr	3-4	19.1	17.1	44.1
Tencent_pr	3-4	19.0	16.8	44.3
GMU_contr	5	17.7	16.5	43.2
GMU_pr	6	17.1	15.9	42.7
CapeTown_pr	7	15.4	14.4	41.2
DENTRA_pr	8	3.5	1.8	24.8

Table 72: Results in ssw-tsn, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	20.5	16.5	46.5
ByteDance_pr	1-2	20.5	16.5	46.5
Tencent_pr	3	16.9	13.9	44.3
GMU_contr	4-5	16.2	13.5	44.0
Tencent_contr	4-5	16.2	13.6	44.1
GMU_pr	6-7	15.4	13.0	43.8
CapeTown_pr	6-7	15.1	13.2	41.9
DENTRA_pr	8	4.1	2.4	24.6

Table 73: Results in tsn-tso, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-3	21.0	18.3	45.7
ByteDance_pr	1-3	20.9	18.4	45.9
Tencent_contr	1-3	20.3	17.6	44.6
Tencent_pr	4-5	19.7	17.4	44.4
GMU_pr	4-5	19.2	17.9	44.5
GMU_contr	6	18.9	17.8	44.4
CapeTown_pr	7	12.0	13.1	38.7
DENTRA_pr	8	5.6	3.6	26.2

Table 74: Results in tso-nso, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-3	17.9	8.8	46.0
ByteDance_pr	1-3	17.8	8.5	46.1
Tencent_contr	1-3	17.5	8.4	46.8
Tencent_pr	4	16.8	8.5	46.1
GMU_contr	5-6	16.0	8.5	46.2
GMU_pr	5-6	15.9	8.7	46.3
CapeTown_pr	7	13.7	6.6	42.7
DENTRA_pr	8	3.0	1.4	24.3

Table 75: Results in nso-xho, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	24.4	9.1	35.9
ByteDance_contr	2	23.9	8.6	35.5
Tencent_pr	3-5	18.6	6.0	31.8
GMU_contr	3-5	18.5	5.9	32.0
GMU_pr	3-5	18.3	5.8	32.0
Tencent_contr	6	18.2	5.5	31.6
DENTRA_pr	7	3.1	1.3	10.1

Table 76: Results in swh-amh, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	26.6	21.8	52.8
ByteDance_pr	1-2	26.5	21.8	52.7
GMU_pr	3-5	21.9	18.6	49.5
GMU_contr	3-5	21.7	18.5	49.5
Tencent_contr	3-5	21.6	18.3	49.7
Tencent_pr	6	21.0	17.7	49.0
DENTRA_pr	7	11.7	10.0	38.8

Table 77: Results in amh-swh, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	9.3	3.9	37.2
ByteDance_pr	1-2	9.3	3.9	37.2
Tencent_pr	3	1.9	0.9	23.1
Tencent_contr	4	1.7	0.8	22.1
GMU_pr	5-7	1.3	0.7	18.5
GMU_contr	5-7	1.3	0.7	18.5
DENTRA_pr	5-7	1.2	0.8	16.3

Table 78: Results in luo-orm, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-2	18.9	6.7	29.9
ByteDance_contr	1-2	18.7	6.6	29.6
Tencent_pr	3-4	13.7	4.4	25.7
GMU_pr	3-4	13.3	4.1	25.7
GMU_contr	5-6	13.2	4.1	25.5
Tencent_contr	5-6	13.2	4.3	25.3
DENTRA_pr	7	0.6	0.4	1.8

Table 79: Results in som-amh, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-2	12.9	8.5	39.7
ByteDance_contr	1-2	12.8	8.6	39.5
Tencent_contr	3-4	8.5	5.5	35.3
Tencent_pr	3-4	8.5	5.5	35.4
GMU_pr	5-6	7.9	5.4	33.5
GMU_contr	5-6	7.8	5.4	33.2
DENTRA_pr	7	1.3	0.9	21.8

Table 80: Results in orm-som, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	15.0	11.3	41.4
ByteDance_contr	2	14.7	11.2	40.9
GMU_pr	3-4	8.8	6.6	31.9
GMU_contr	3-4	8.7	6.5	31.9
Tencent_pr	5-6	5.3	3.8	26.2
Tencent_contr	5-6	5.2	3.8	25.6
DENTRA_pr	7	3.9	2.5	21.7

Table 81: Results in swh-luo, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	12.0	8.8	38.4
ByteDance_pr	1-2	12.0	8.7	38.6
GMU_contr	3	6.4	5.0	29.5
GMU_pr	4	6.1	4.7	29.1
DENTRA_pr	5	3.1	2.8	24.7
Tencent_contr	6-7	2.8	2.0	21.2
Tencent_pr	6-7	2.8	2.1	21.4

Table 82: Results in amh-luo, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	12.7	8.5	39.0
ByteDance_pr	1-2	12.6	8.5	39.0
GMU_contr	3-5	9.1	6.3	34.7
Tencent_contr	3-5	9.0	6.2	35.8
GMU_pr	3-5	8.9	6.1	34.2
Tencent_pr	6	8.5	5.6	34.8
DENTRA_pr	7	4.1	2.8	20.4

Table 83: Results in luo-som, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
Tencent_contr	1-4	17.9	14.6	39.7
ByteDance_contr	1-4	17.8	14.9	39.8
Tencent_pr	1-4	17.8	14.7	39.4
ByteDance_pr	1-4	17.7	14.9	39.8
GMU_pr	5-6	15.7	13.5	37.5
GMU_contr	5-6	15.7	13.4	37.4
DENTRA_pr	7	4.2	2.7	19.2

Table 84: Results in hau-ibo, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	10.0	4.6	25.5
ByteDance_contr	2	9.7	4.5	25.6
Tencent_pr	3-4	5.2	2.9	21.4
Tencent_contr	3-4	5.1	2.8	20.8
GMU_pr	5	4.0	2.5	20.5
GMU_contr	6	3.9	2.5	20.3
DENTRA_pr	7	2.3	1.2	13.8

Table 85: Results in ibo-yor, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	4.0	2.7	22.3
ByteDance_pr	1-2	4.0	2.7	22.4
DENTRA_pr	3	1.9	0.9	13.7
Tencent_pr	4-7	0.5	0.2	13.9
GMU_pr	4-7	0.4	0.3	13.6
GMU_contr	4-7	0.4	0.3	13.5
Tencent_contr	4-7	0.3	0.2	14.0

Table 86: Results in yor-fuv, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-2	9.5	7.3	32.1
ByteDance_contr	1-2	9.4	7.2	31.9
Tencent_contr	3-4	3.4	2.8	22.4
Tencent_pr	3-4	3.4	2.9	23.1
GMU_pr	5-7	3.1	2.5	18.8
DENTRA_pr	5-7	3.1	1.7	19.5
GMU_contr	5-7	3.1	2.7	19.6

Table 87: Results in fuv-hau, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	21.6	18.8	46.7
ByteDance_pr	1-2	21.5	18.7	46.3
Tencent_pr	3-4	17.8	15.6	44.2
Tencent_contr	3-4	17.1	15.0	44.1
GMU_pr	5	17.0	14.9	42.6
GMU_contr	6	16.7	14.7	42.3
DENTRA_pr	7	3.8	2.2	20.0

Table 88: Results in ibo-hau, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1	13.5	10.9	35.0
ByteDance_pr	2	13.2	10.6	34.7
Tencent_contr	3	12.6	9.5	33.1
Tencent_pr	4	12.2	9.3	33.1
GMU_pr	5-6	11.5	9.3	32.2
GMU_contr	5-6	11.5	9.3	32.3
DENTRA_pr	7	2.4	1.0	13.1

Table 89: Results in yor-ibo, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-2	5.7	2.3	20.1
ByteDance_contr	1-2	5.6	2.2	19.8
DENTRA_pr	3	2.0	1.2	13.4
GMU_pr	4-5	1.0	0.6	9.2
Tencent_pr	4-5	0.9	0.4	10.2
Tencent_contr	6-7	0.8	0.4	9.4
GMU_contr	6-7	0.7	0.4	8.0

Table 90: Results in fuv-yor, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	5.0	3.5	23.8
ByteDance_contr	2	4.7	3.4	23.5
DENTRA_pr	3	3.0	1.5	21.1
GMU_pr	4	0.4	0.3	13.9
GMU_contr	5-7	0.4	0.3	13.6
Tencent_contr	5-7	0.4	0.2	14.0
Tencent_pr	5-7	0.4	0.1	14.0

Table 91: Results in hau-fuv, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-2	14.6	12.1	38.4
ByteDance_contr	1-2	14.4	11.9	38.1
Tencent_contr	3	9.1	7.6	34.1
Tencent_pr	4	8.5	7.2	32.8
GMU_pr	5	7.2	5.8	25.4
GMU_contr	6	6.9	5.7	26.4
DENTRA_pr	7	3.8	2.3	20.3

Table 92: Results in wol-hau, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	8.6	5.9	27.5
ByteDance_contr	2	8.2	5.6	27.1
Tencent_pr	3-4	3.8	3.1	20.0
GMU_pr	3-4	3.7	2.4	15.7
GMU_contr	5-6	3.5	2.3	14.9
Tencent_contr	5-6	3.4	2.7	18.8
DENTRA_pr	7	3.2	1.6	19.2

Table 93: Results in hau-wol, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	5.2	3.4	22.3
ByteDance_contr	2	5.0	3.3	21.9
DENTRA_pr	3	2.5	1.4	18.4
GMU_pr	4	1.3	0.9	10.7
GMU_contr	5	1.1	0.8	10.4
Tencent_pr	6-7	1.0	0.9	10.8
Tencent_contr	6-7	0.9	0.9	10.6

Table 94: Results in fuv-wol, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	4.4	3.0	23.0
ByteDance_contr	2	4.2	2.8	22.5
DENTRA_pr	3	2.5	1.3	19.3
GMU_pr	4-7	0.4	0.3	14.3
GMU_contr	4-7	0.4	0.3	13.9
Tencent_pr	4-7	0.4	0.3	14.4
Tencent_contr	4-7	0.3	0.2	14.6

Table 95: Results in wol-fuv, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	27.9	23.5	52.9
ByteDance_contr	2	27.5	23.1	52.7
Tencent_contr	3-4	24.2	20.8	50.7
Tencent_pr	3-4	24.2	20.8	50.4
GMU_contr	5-6	22.6	19.3	49.4
GMU_pr	5-6	22.4	19.3	49.0
DENTRA_pr	7	4.5	3.7	25.2

Table 96: Results in kin-swh, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-2	14.9	11.9	41.2
ByteDance_contr	1-2	14.8	11.8	41.0
Tencent_pr	3-4	9.1	7.9	32.7
Tencent_contr	3-4	9.0	7.8	32.9
GMU_contr	5-6	7.2	5.7	32.5
GMU_pr	5-6	7.0	5.5	32.0
DENTRA_pr	7	3.2	1.7	22.9

Table 97: Results in lug-lin, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-2	18.5	13.3	44.2
ByteDance_contr	1-2	18.3	13.2	44.1
Tencent_contr	3	15.8	11.9	41.9
Tencent_pr	4	15.0	10.9	41.5
GMU_contr	5	12.4	9.4	39.0
GMU_pr	6	11.9	9.0	38.5
DENTRA_pr	7	3.7	2.5	21.9

Table 98: Results in nya-kin, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	13.0	7.9	43.3
ByteDance_pr	1-2	13.0	8.0	43.6
Tencent_pr	3	8.4	5.9	37.1
Tencent_contr	4-5	7.5	5.6	35.9
GMU_pr	4-5	7.2	5.5	34.5
GMU_contr	6	6.5	5.0	33.6
DENTRA_pr	7	3.0	1.9	21.9

Table 99: Results in swh-lug, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	13.8	10.4	42.2
ByteDance_pr	1-2	13.7	10.1	42.0
Tencent_pr	3-4	11.6	8.6	39.4
Tencent_contr	3-4	11.5	8.8	38.9
GMU_pr	5-6	10.6	8.1	39.8
GMU_contr	5-6	10.5	8.0	39.4
DENTRA_pr	7	4.4	2.3	26.0

Table 100: Results in lin-nya, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	17.7	13.6	43.1
ByteDance_pr	1-2	17.6	13.6	42.9
Tencent_pr	3-5	11.2	9.0	34.8
GMU_contr	3-5	11.0	8.4	36.9
GMU_pr	3-5	10.5	7.9	36.5
Tencent_contr	6	10.4	8.8	33.8
DENTRA_pr	7	4.0	2.4	22.4

Table 101: Results in lin-kin, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	12.1	7.4	42.0
ByteDance_pr	1-2	12.0	7.2	42.0
Tencent_contr	3-4	7.1	4.8	35.2
Tencent_pr	3-4	7.1	5.0	34.8
GMU_pr	5-7	3.2	2.0	22.4
DENTRA_pr	5-7	3.2	1.8	23.9
GMU_contr	5-7	3.2	1.9	22.5

Table 102: Results in kin-lug, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	23.0	18.6	48.4
ByteDance_pr	1-2	23.0	18.6	48.6
GMU_pr	3-5	20.7	17.2	47.8
GMU_contr	3-5	20.7	17.2	47.8
Tencent_contr	3-5	20.5	17.1	47.1
Tencent_pr	6	20.3	16.9	47.1
DENTRA_pr	7	4.7	3.1	26.4

Table 103: Results in nya-swh, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	18.6	9.0	47.6
ByteDance_pr	1-2	18.6	9.1	47.6
GMU_contr	3-4	15.8	7.3	46.4
GMU_pr	3-4	15.7	7.4	46.3
Tencent_pr	5-6	14.8	6.8	45.7
Tencent_contr	5-6	14.7	6.8	45.3
DENTRA_pr	7	7.1	3.5	35.5

Table 104: Results in amh-zul, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-2	17.9	14.7	43.4
ByteDance_contr	1-2	17.7	14.5	43.3
Tencent_contr	3-5	14.1	11.6	40.4
GMU_contr	3-5	14.0	11.6	40.2
Tencent_pr	3-5	14.0	11.6	40.5
GMU_pr	6	13.7	11.5	39.7
DENTRA_pr	7	6.7	5.0	29.3

Table 105: Results in yor-swh, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	11.1	4.9	26.9
ByteDance_contr	2	10.9	4.8	26.6
Tencent_pr	3-4	5.2	3.2	21.8
Tencent_contr	3-4	5.0	3.0	21.5
GMU_contr	5	3.9	2.8	20.9
GMU_pr	6	3.7	2.7	20.8
DENTRA_pr	7	2.3	1.5	14.8

Table 106: Results in swh-yor, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	21.2	7.1	32.3
ByteDance_pr	1-2	21.1	7.2	32.2
GMU_pr	3	16.6	5.1	29.3
GMU_contr	4-5	16.4	5.0	29.3
Tencent_pr	4-5	16.4	5.3	28.8
Tencent_contr	6	15.5	4.9	28.1
DENTRA_pr	7	3.1	1.2	11.4

Table 107: Results in zul-amh, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	22.9	19.9	48.0
ByteDance_contr	2	22.6	19.6	47.9
Tencent_pr	3-4	20.0	17.5	45.3
Tencent_contr	3-4	19.8	17.1	45.9
GMU_pr	5-6	18.9	16.7	44.0
GMU_contr	5-6	18.7	16.5	43.9
DENTRA_pr	7	4.2	2.6	22.7

Table 108: Results in kin-hau, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	21.5	16.1	47.1
ByteDance_pr	1-2	21.4	16.0	47.1
Tencent_contr	3-4	18.0	13.6	44.0
Tencent_pr	3-4	17.8	13.3	43.5
GMU_pr	5	14.3	11.1	40.8
GMU_contr	6	14.2	10.9	40.6
DENTRA_pr	7	3.7	1.9	20.2

Table 109: Results in hau-kin, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	13.9	9.1	40.9
ByteDance_contr	2	13.7	9.1	40.7
GMU_contr	3	12.6	8.1	40.3
GMU_pr	4-6	12.4	8.0	40.3
Tencent_pr	4-6	12.4	8.0	40.5
Tencent_contr	4-6	12.3	8.1	40.0
DENTRA_pr	7	5.6	4.0	27.5

Table 110: Results in nya-som, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_contr	1-2	14.4	10.7	43.1
ByteDance_pr	1-2	14.4	10.7	43.4
Tencent_contr	3	13.3	10.1	42.5
Tencent_pr	4	13.1	9.9	42.4
GMU_pr	5-6	12.8	9.7	42.5
GMU_contr	5-6	12.8	9.7	42.4
DENTRA_pr	7	6.2	4.6	30.3

Table 111: Results in som-nya, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-2	12.1	7.3	41.2
ByteDance_contr	1-2	12.0	7.3	41.2
Tencent_pr	3	7.3	5.2	35.2
Tencent_contr	4	7.0	5.2	34.7
GMU_pr	5-6	5.7	4.4	31.4
GMU_contr	5-6	5.5	4.2	31.0
DENTRA_pr	7	2.3	1.5	23.2

Table 112: Results in xho-lug, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1-2	13.0	6.1	40.0
ByteDance_contr	1-2	12.9	6.1	39.8
Tencent_contr	3-4	11.2	5.0	37.6
Tencent_pr	3-4	11.1	5.1	37.8
GMU_pr	5-6	9.6	4.6	36.0
GMU_contr	5-6	9.5	4.7	36.0
DENTRA_pr	7	2.3	1.4	22.9

Table 113: Results in lug-xho, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	16.5	13.5	40.1
ByteDance_contr	2	15.9	13.0	39.6
Tencent_contr	3-4	11.7	9.0	36.3
Tencent_pr	3-4	11.6	9.1	36.5
GMU_contr	5	8.7	6.9	31.1
GMU_pr	6	8.3	6.6	30.0
DENTRA_pr	7	5.6	4.3	26.1

Table 114: Results in wol-swh, sorted by spBLEU.

System	Rank	spBLEU	BLEU	chrF2
ByteDance_pr	1	8.5	5.9	28.5
ByteDance_contr	2	8.3	5.9	28.0
GMU_pr	3-5	3.5	2.5	15.9
GMU_contr	3-5	3.4	2.3	15.0
Tencent_pr	3-5	3.4	2.7	19.1
DENTRA_pr	6-7	3.0	1.8	18.8
Tencent_contr	6-7	3.0	2.4	18.0

Table 115: Results in swh-wol, sorted by spBLEU.

Findings of the WMT 2022 Shared Tasks in Unsupervised MT and Very Low Resource Supervised MT

Marion Weller-Di Marco and Alexander Fraser
Center for Information and Language Processing
LMU Munich
`{dimarco,fraser}@cis.uni-muenchen.de`

Abstract

We present the findings of the WMT2022 Shared Tasks in Unsupervised MT and Very Low Resource Supervised MT with experiments on the language pairs German to/from Upper Sorbian, German to/from Lower Sorbian and Lower Sorbian to/from Upper Sorbian. Upper and Lower Sorbian are minority languages spoken in the Eastern parts of Germany. There are active language communities working on the preservation of the languages who also made the data used in this Shared Task available.

In total, four teams participated on this Shared Task, with submissions from three teams for the unsupervised sub task, and submissions from all four teams for the supervised sub task. In this overview paper, we present and discuss the results.

1 Introduction

For a large majority of the world's languages, only limited resources are available to train and provide NLP tools. The need for parallel data in a (supervised) translation scenario aggravates this problem further. The Shared Tasks in Unsupervised MT and Very Low Resource Supervised MT aim at promoting the research on translating low and very low resourced languages.

Following the Shared Tasks in the two previous years (Libovický and Fraser, 2021; Fraser, 2020), we continue to cooperate with the Sorbian community, namely the Sorbian Institute[1] and the Witaj Sprachzentrum (Witaj Language Center)[2] for this year's Shared Task. We offer all translation directions between the languages Upper Sorbian, Lower Sorbian and German, for both supervised and unsupervised translation.

Upper and Lower Sorbian are minority languages spoken in the eastern part of Germany in the federal states of Saxony and Brandenburg. With only 30k and 7k native speakers, there are only few resources available. However, as western Slavic languages, Upper and Lower Sorbian are closely related to Polish and Czech and can thus make use of the comparatively large data sets available for those languages.

In this year, four teams participated in the Shared Task, resulting in three to four submissions for each language pair for both the supervised and unsupervised variants.

2 Tasks and Evaluation

In contrast to the previous Shared Tasks, all language combinations between Upper Sorbian, Lower Sorbian and German are considered, resulting in the six following translation pairs:

- Upper Sorbian ↔ German
- Lower Sorbian ↔ German
- Upper Sorbian ↔ Lower Sorbian

Factoring in the variants supervised and unsupervised translation for each language pair, there is a total of 12 translation pairs.

For the evaluation, we follow the strategy employed in the previous Shared Task and use BLEU scores (Papineni et al., 2002) and chrF scores (Popović, 2015) as implemented in sacreBLEU (Post, 2018).[3] Furthermore, we evaluate the submissions using BERTScore (Zhang et al., 2020)[4] with XLM-RoBERTa Large (Conneau et al., 2020) as an underlying model for translations into German[5].

[1] https://www.serbski-institut.de/en/Institute/

[2] https://www.witaj-sprachzentrum.de/

[3] BLEU score signature: `nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.2.0`
chrf2 score signature: `nrefs:1|case:mixed|eff:yes|nc:6|nw:0|space:no|version:2.2.0`

[4] https://github.com/Tiiiger/bert_score

[5] BERTScore signatures: `xlm-roberta-large_L17_no-idf_version=0.3.11(hug_trans=4.22.2)_fast-tokenizer` and `xlm-roberta-large_L17_idf_version=0.3.11(hug_trans=4.22.2)_fast-tokenizer`

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 801 - 805
December 7-8, 2022 ©2022 Association for Computational Linguistics

HSB ↔ DE	449.057	parallel sentences
DSB ↔ DE	40.193	parallel sentences
DSB ↔ HSB	62.564	parallel sentences

DSB	220.419	monolingual sentences
HSB	1.132.850	monolingual sentences

Table 1: Training data per language pair. The data sets have been made available by the Sorbian Institute (monolingual data) and The Witaj Sprachzentrum (both parallel and monolingual data).

We decided against using COMET Scores (Rei et al., 2020). This metric considers both the source language and the target language, but because it relies on XLM-R models, it does not support the Sorbian languages.

3 Data

To allow for a direct comparison between the different submissions, we only allowed training based on the resources released for the task, as well as resources for related languages (German, Czech and Polish data from the WMT news tasks[6] or available in the OPUS project[7]). In particular, the use of large models pre-trained on large data sets was not allowed. Table 1 gives an overview of the parallel and monolingual training data for the Sorbian languages.

For the unsupervised translation sub-task, we restricted the the data set as follows: all released Upper/Lower Sorbian data could be used, with the exception of the parallel Upper Sorbian ↔ Lower Sorbian corpus. Furthermore, the German side of the parallel German ↔ Upper Sorbian and German ↔ Lower Sorbian training corpora was excluded. This setup allowed us to make maximum use of the low-resourced languages without providing parallel data.

4 Submitted Systems

Four teams participated in the supervised sub-task[8], and three teams participated in the unsupervised sub-task. We present a brief system description of each team's submission, with an overview of the results listed in tables 2 to 7. Table 8 gives a brief overview of some relevant details; for more detailed information, we refer the reader to the respective system description papers.

AIC (Shapiro et al., 2022) For the unsupervised system, they trained an unsupervised phrase-based statistical machine translation (UPBSMT) system on each pair independently. They trained a De-Slavic mBART model from Scratch (Random initialization) on the following languages: Polish (pl), Czech (cs), German (de), Upper Sorbian (hsb), and Lower Sorbian (dsb). They then fine-tuned their mBART on the synthetic parallel data generated by the UPBSMT model along with authentic parallel data (de ↔ pl, de ↔ cs). They further fine-tuned their unsupervised system on authentic parallel data (hsb ↔ dsb, de ↔ dsb, de ↔ hsb) to submit the supervised low-resource system.

MUNI NLP (Signoroni and Rychlý, 2022) This team submitted supervised NMT systems for the Lower Sorbian-German and Lower Sorbian-Upper Sorbian language pairs, in both translation directions. They employed a new subword tokenization algorithm, High Frequency Tokenizer (HFT), to obtain more meaningful subword vocabularies. They tested this against BPE in the first round of experiments where they trained two different models on the data tokenized with each tokenizer, so four systems in total: two standard Transformers and two Transformers with hyperparameters optimized for the dataset size. They then followed the Data Diversification procedure (Nguyen et al., 2020) generating and collating authentic and synthetic data alternatively from each previous system and the original parallel data to create an augmented dataset. Then, they trained a Transformer model on these new data, tokenized with HFT, to obtain the final system. Thus, the approach is based only on the original parallel corpus.

Huawei TSC (Li et al., 2022) Huawei Translation Services Center participated in all 6 supervised tracks. Their systems are build on deep Transformer models with a large filter size. First, they selected a base multilingual model with German-Czech (DE-CS) and German-Polish (DE-PL) parallel data for all of the 6 tracks. They then utilized regularized dropout (R-Drop), back translation, fine-tuning and ensemble multilingual models to improve on the best individual system performance. For the unsupervised task submission, they applied their pre-trained multilingual system with zero-shot.

[6] https://www.statmt.org/wmt22/translation-task.html

[7] https://opus.nlpl.eu/

[8] There were submissions by a fifth team in for the supervised task. We do not have system descriptions for this team's submissions, and thus listed their results separately in table 9.

	DE-DSB	
System	BLEU	chrF2
HuaweiTSC	73.9	87.1
MUNI-NLP	50.5	74.1
AIC	48.2	73.0
PICT-NLP	20.8	44.1

	DSB-DE			
System	BLEU	chrF2	BERT$_F$	BERT$_{F\_IDF}$
HuaweiTSC	62.5	80.9	0.9792	0.9764
MUNI-NLP	49.5	73.0	0.9664	0.9613
AIC	39.4	66.2	0.9542	0.9463
PICT-NLP	25.4	51.3	0.9246	0.9125

Table 2: Results for supervised DE-DSB and DSB-DE translation.

	DE-HSB	
System	BLEU	chrF2
HuaweiTSC	70.7	85.5
AIC	51.0	73.7
PICT-NLP	25.7	49.1

	HSB-DE			
System	BLEU	chrF2	BERT$_F$	BERT$_{F\_IDF}$
HuaweiTSC	71.9	85.3	0.9843	0.9825
AIC	47.5	71.4	0.9637	0.9574
PICT-NLP	29.7	53.8	0.9317	0.9207

Table 3: Results for supervised DE-HSB and HSB-DE translation.

	DSB-HSB	
System	BLEU	chrF2
HuaweiTSC	86.8	94.0
MUNI-NLP	72.2	87.5
AIC	65.8	83.9
PICT-NLP	49.1	65.5

	HSB-DSB	
System	BLEU	chrF2
HuaweiTSC	88.0	94.4
MUNI-NLP	72.3	87.5
AIC	66.6	84.3
PICT-NLP	50.7	66.9

Table 4: Results for supervised DSB-HSB and HSB-DSB translation.

	DE-DSB	
System	BLEU	chrF2
HuaweiTSC	9.0	32.6
AIC	1.2	22.1
PICT-NLP	0.2	8.1

	DSB-DE			
System	BLEU	chrF2	BERT$_F$	BERT$_{F\_IDF}$
HuaweiTSC	11.5	33.9	0.9141	0.8970
AIC	4.0	26.9	0.8567	0.8434
PICT-NLP	0.0	5.0	0.7822	0.7693

Table 5: Results for unsupervised DE-DSB and DSB-DE translation.

	DE-HSB	
System	BLEU	chrF2
AIC	17.9	48.5
HuaweiTSC	10.4	33.4
PICT-NLP	0.5	14.3

	HSB-DE			
System	BLEU	chrF2	BERT$_F$	BERT$_{F\_IDF}$
AIC	18.0	46.9	0.9046	0.8937
HuaweiTSC	13.5	35.8	0.9162	0.8996
PICT-NLP	0.3	13.6	0.8306	0.8194

Table 6: Results for unsupervised DE-HSB and HSB-DE translation.

	DSB-HSB	
System	BLEU	chrF2
AIC	44.2	72.9
HuaweiTSC	–	–
PICT-NLP	10.4	48.6

	HSB-DSB	
System	BLEU	chrF2
AIC	35.9	67.4
PICT-NLP	9.3	44.2
HuaweiTSC	2.4	16.1

Table 7: Results for unsupervised DSB-HSB and HSB-DSB translation.

team	**data** (in addition to the provided de/hsb/dsb corpora)	**synthetic data/ back translation**	**segmentation (vocab. size)**	**toolkits**
AIC	DE (431.4M), CS (111.1M) PL (13.4M), PL-DE (12.4M) CS-DE (12.4M)	synthetic data through UPBSMT	SentencePiece (32k)	Fairseq
HUAWEI	DE-CS (55.9M), DE-PL (66.5M), DE (20M)	back-translation with sampling (Graça et al., 2019)	SentencePiece (40k)	Fairseq Marian
MUNI	–	Data diversification (Nguyen et al., 2020)	High Frequency Tokenizer (4k)	Fairseq
PICT	DE (53.3k)	–	BPE	Fairseq Facebook's XLM

Table 8: Overview of methods and data.

PICT NLP (Vyawahare et al., 2022) They implemented the XLM's Masked Language Model (MLM) for unsupervised learning. They trained it only using the monolingual data provided by the organizers and the OPUS project. Finally, they also applied XLM preprocessing to the data before training.

For supervised learning, they trained language models such as LSTM and attention based transformer models with the help of the Fairseq library. They trained it using monolingual data provided by the organizers. They applied the inbuilt tokenization provided by Fairseq on the data.

5 System Results

Tables 2 to 7 list the results of the submitted systems in terms of BLEU and chrF2 for all systems, and additionally BERT scores for those translating into German. For the BERT scores, we list both BERT_F and BERT_F with idf weighting to give less weight to commonly occuring words. The ordering of the systems is consistent across all metrics.

The supervised systems obtain higher results than the unsupervised systems. The language pair DSB ↔ HSB obtained comparatively high scores for both supervised and unsupervised translation which is very probably due to the high similarity between the two languages.

Overall, we see no winner across all tasks: HuaweiTSC has the best scores across all supervised translation tasks, followed by MUNI-NLP for the DE-to/from-DSB and DSB-to/from-HSB translations. These two language pairs only have comparably small parallel data sets which are, notably, the sole basis of MUNI-NLP's submissions.

For the unsupervised translation (where MUNI-NLP did not participate), AIC has the strongest results with the exception of DSB-to/from-DE, where HuaweiTSC is leading.

6 Conclusion

In the WMT 2022 Shared Task on Unsupervised and Very Low Resource MT, we provided the participants with resources for all possible translation directions for the three languages Upper Sorbian, Lower Sorbian and German, of which Upper Sorbian ↔ Lower Sorbian is a new language pair in comparison to last year's shared task.

The participating teams submitted strong systems relying on a wide range of methods. Using modeling techniques such as pre-training on parallel data of related languages is important, as is the creation of synthetic data for which we saw the application of different methods. However we also saw that careful modeling on a small data set only can lead to good results.

We hope that this Shared Task will continue to increase the interest in research on methods for under-resourced languages, both for supervised and unsupervised approaches.

Acknowledgements

This work was supported by the DFG (grant FR 2829/4-1). We would also like to thank Marko Měškank and Olaf Langner (Witaj Sprachzentrum) and Hauke Bartels and Marcin Szczepański (Sorbian Institute).

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexander Fraser. 2020. Findings of the WMT 2020 shared tasks in unsupervised MT and very low resource supervised MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 765–771, Online. Association for Computational Linguistics.

Miguel Graça, Yunsu Kim, Julian Schamper, Shahram Khadivi, and Hermann Ney. 2019. Generalizing back-translation in neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 45–52, Florence, Italy. Association for Computational Linguistics.

Shaojun Li, Yuanchang Luo, Daimeng Wei, Zongyao Li, Hengchao Shang, Xiaoyu Chen, Zhanglin Wu, Jinlong Yang, Zhiqiang Rao, Zhengzhe Yu, Yuhao Xie, Lizhi Lei, Hao Yang, and Ying Qin. 2022. HW-TSC Systems for WMT22 Very Low Resource Supervised MT Task. In *Proceedings of the Seventh Conference on Machine Translation*. Association for Computational Linguistics.

Jindřich Libovický and Alexander Fraser. 2021. Findings of the WMT 2021 shared tasks in unsupervised MT and very low resource supervised MT. In *Proceedings of the Sixth Conference on Machine Translation*, pages 726–732, Online. Association for Computational Linguistics.

Xuan-Phi Nguyen, Shafiq R. Joty, Kui Wu, and Ai Ti Aw. 2020. Data diversification: A simple strategy for neural machine translation. In *NeurIPS*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ahmad Shapiro, Mahmoud Tarek Salama, Omar Khaled Abdelhakim, Mohamed Essam Fayed, Ayman Khalafallah, and Noha Adly. 2022. The AIC System for the WMT 2022 Unsupervised MT and Very Low Resource Supervised MT Task. In *Proceedings of the Seventh Conference on Machine Translation*. Association for Computational Linguistics.

Edoardo Signoroni and Pavel Rychlý. 2022. MUNI-NLP Systems for Lower Sorbian-German and Lower Sorbian-Upper Sorbian Machine Translation @ WMT22. In *Proceedings of the Seventh Conference on Machine Translation*. Association for Computational Linguistics.

Aditya Vyawahare, Rahul Tangsali, Aditya Mandke, Onkar Litake, and Dipali Kadam. 2022. PICT-NLP@WMT22-EMNLP2022: Unsupervised and Very-Low Resource Supervised Translation on German and Sorbian Variant Languages. In *Proceedings of the Seventh Conference on Machine Translation*. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

A Further Results

Table 9 lists the results of another submission that did not provide further details.

	BLEU	chrF2	BERT$_F$	BERT$_{F\_IDF}$
DE-DSB	58.2	79.5	–	–
DSB-DE	61.5	80.4	0.9784	0.9755
DE-HSB	67.3	83.9	–	–
HSB-DE	71.2	85.1	0.9840	0.9821
DSB-HSB	72.8	87.7	–	–
HSB-DSB	72.2	87.6	–	–

Table 9: Results for supervised translation of a team that we were not able to contact.

Overview and Results of MixMT Shared-Task at WMT 2022

Vivek Srivastava
TCS Research
Pune, Maharashtra, India
srivastava.vivek2@tcs.com

Mayank Singh
IIT Gandhinagar
Gandhinagar, Gujarat, India
singh.mayank@iitgn.ac.in

Abstract

In this paper, we present an overview of the WMT 2022 shared task on code-mixed machine translation (MixMT). In this shared task, we hosted two code-mixed machine translation subtasks in the following settings: (i) monolingual to code-mixed translation and (ii) code-mixed to monolingual translation. In both the subtasks, we received registration and participation from teams across the globe showing an interest and need to immediately address the challenges with machine translation involving code-mixed and low-resource languages.

1 Introduction

Code-mixing (or code-switching) is an interesting manifestation of multilingualism in communities across the globe. Lately, we observe an uptick in the interest and efforts of the computational linguistic community to solve a multitude of challenges with code-mixed languages. Several interesting resources and computational models have been proposed for problems such as language identification (Barman et al., 2014; Thara and Poornachandran, 2021), text generation (Gupta et al., 2020; Rizvi et al., 2021), and sentiment analysis (Chakravarthi et al., 2020; Patwa et al., 2020).

Machine translation which is an active area of research and development for monolingual languages is at the outset for code-mixed languages (Chen et al., 2022). In this shared task, we aim to explore the machine translation task involving a popular code-mixed language i.e., Hinglish (code-mixing of Hindi and English). Through both subtasks, we aim to address the challenges in building a real-world multilingual translation system involving code-mixed language as the source/target.

Similar to the recent events on code-mixed languages (Chen et al., 2022; Srivastava and Singh, 2021b; Patwa et al., 2020), the MixMT shared task has received participation and engagement with teams from across the globe. In total, we received registration from 38 teams. Throughout the competition, seven teams actively participated and submitted their system for the development phase, test phase, and human evaluation phase.

2 MixMT: Code-mixed Machine Translation

2.1 The two subtasks

In the MixMT shared task, we hosted two subtasks involving a code-mixed language i.e. Hinglish. A brief description of both subtasks is given below:

1. **Monolingual to code-mixed machine translation (M2CM)**: In this subtask, Hindi and English are the two source languages and the target language is Hinglish. The source Hindi and English sentences are translations of each other. The Hindi language sentences are written in the Devanagari script whereas the target Hinglish language text is written in the Roman script.
2. **Code-mixed to monolingual machine translation (CM2M)**: In this subtask, Hinglish is the source language and the target language is English. Both the English and Hinglish text are written in Roman script.

2.2 Dataset

Training datasets: We provide the following training datasets for both subtasks:

1. **M2CM**: For this subtask, HinGE (Srivastava and Singh, 2021a) is the primary training dataset. It contains parallel English-Hindi sentences along with multiple human-generated Hinglish sentences. For each data instance, it also contains two synthetically generated Hinglish sentences. The dataset was also used as part of the HinglishEval shared task (Srivastava and Singh, 2021b). We provide the entire HinglishEval data of $\approx$ 2k samples (train, validation, and test set together) as part of the training data for the MixMT shared task.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 806 - 811
December 7-8, 2022 ©2022 Association for Computational Linguistics

2. **CM2M**: For this subtask, PHINC (Srivastava and Singh, 2020) is the primary training dataset. It contains 13,738 parallel sentences in the Hinglish and English languages.

Evaluation datasets: To evaluate both the subtasks, we have created an in-house hidden evaluation dataset. For both subtasks, the validation dataset contains 500 samples and the test dataset has 1500 samples. The evaluation dataset is available here: `bit.ly/3UZLdFm`.

2.3 Baseline system and evaluation

We use Google Translate as a baseline for both subtasks. For *M2CM* subtask, we translate Hindi sentences (in Devanagari script) into English and evaluate them against the reference Hinglish sentences. For *CM2M* subtask, we translate the Hinglish sentences into English by setting the language of the Hinglish text as Hindi.

Evaluation: We use two evaluation metrics for both the subtasks: ROUGE-L (F1-score) and Word Error Rate (WER). Also, we perform a human-based qualitative evaluation of both subtasks. Table 1 shows the policy of the human-based evaluation of both subtasks.

2.4 Constrained system

We distinguish between the constrained and unconstrained systems based on the following criteria:

1. The system using an external dataset (apart from HinGE and PHINC datasets) will be considered unconstrained.
2. We allow public pre-trained models in a constrained system given that it is accessible to all the teams.

3 Submissions

We received the submissions from seven teams (listed alphabetically):

1. **CNLP-NITS-PP** (Laskar et al., 2022): They leverage the external parallel corpus (Samanantar (Ramesh et al., 2022)) to train their translation model which is built using OpenNMT-py framework (Klein et al., 2017) with the default setting. To generate the synthetic dataset, they transliterate and align the words in parallel sentences. Finally, they augment the provided dataset with the synthetic dataset to train their model.
2. **Domain Curricula for Code-switched MT (DC)** (Raheem and Elrashid, 2022): They experiment with different combinations of pre-training fine-tuning setups. They leverage the synthetic code-mixed dataset generated using the IIT-B parallel corpus (Kunchukuttan et al., 2018) and matrix language theory (Myers-Scotton). Further, the mixed data pretraining with synthetic and task-specific data shows the best result on the evaluation dataset. To build the translation model, they use transformers (Vaswani et al., 2017) and fairseq toolkit (Ott et al., 2019).
3. **Gui** (Gahoi et al., 2022): They leverage the multilingual pre-trained models to build their translation system. For *M2CM* task, they fine-tune multilingual-BART (Liu et al., 2020) on the task-specific data with reduced vocabulary. They reduce the vocabulary using the tokens present in the task dataset, IIT-B parallel corpus (Kunchukuttan et al., 2018), and the Dakshina dataset (Roark et al., 2020). They also perform the post-processing on the output generated from the fine-tuned model. For *CM2M* task, they finetune Salesken.AI's pre-trained model provided on Huggingface Transformers which is a finetuned Helsinki's OPUS-MT model on AI4Bharat's Samanantar dataset (Ramesh et al., 2022).
4. **MUCS** (Hegde and Shashirekha, 2022): Their translation model for both the task is built around transliteration (Bhat et al., 2015) and fine-tuning the IndicTrans pre-trained model (Ramesh et al., 2022). They generate synthetic parallel data using the Samanantar corpus (Ramesh et al., 2022). They further fine-tune the IndicTrans model jointly with the synthetic and task-specific datasets.
5. **NICT** (Dabre, 2022): They propose a synthetic code-mixed data based pre-training and a multi-way fine-tuning strategy. To generate the synthetic dataset, they leverage the Samanantar corpus (Ramesh et al., 2022), the transliteration toolkit[1], and a min-max based approach for word alignment (Zenkel et al., 2021). They pre-train a multilingual model on the synthetic Hinglish-English and English-synthetic Hinglish dataset. To perform the multi-way fine-tuning, they fine-tune the pre-trained model on Hinglish to English and English to Hinglish jointly using a small subset of the English side

[1] `https://github.com/anoopkunchukuttan/indic_nlp_library`

Rating	*M2CM*	*CM2M*
5	Best Generated Hinglish sentence	Correctly translated sentence conveying the exact same information as the source sentence
4	A Hinglish sentence with minimal grammtical mistakes but less likely in general parlance	A translated sentence with minimal grammatical mistakes
3	A Hinglish sentence that contains mainly grammtical mistakes	A translated sentence that contains mainly grammatical mistakes
2	A Hinglish sentence containing fairly large volumes of lexical and grammatical mistakes	A translated sentence containing fairly large volumes of lexical and grammatical mistakes
1	Worst Generated Sentence. They are monolingual either in Romanized Hindi or English	A translated sentence with poor semantics and irrelevant to the source sentence

Table 1: Human-based evaluation policy for *M2CM* and *CM2M* subtasks. The underlined phrase highlights the center of attention for the corresponding rating.

of the synthetic data and the entire parallel corpus (PHINC & HinGE) together. They use a denoising strategy similar to BART (Lewis et al., 2020) by randomly masking English words in the source sentence. They use the YANMTT toolkit (Dabre and Sumita, 2021) to their translation model.

6. **SIT-NMT** (Khan et al., 2022): They experiment with a variety of multilingual pre-trained models such as multilingual BART (Liu et al., 2020) and multilingual T5 (Xue et al., 2021). They fine-tune these pre-trained models on external datasets. For *M2CM task*, they use Kaggle Hi-En (Chokhra, 2020) and MUSE Hi-En dictionary (Lample et al., 2018). For *CM2M* task, they use CMU movie reviews data (Zhou et al., 2018) and CALCS'21 dataset (Chen et al., 2022). They also use selected WMT'14 News Hi-En sentences (Bojar et al., 2014) and the MTNT Fr-En and Ja-En data (Michel and Neubig, 2018). In addition, they also increase the size of the dataset by back-translating samples of the English side of Tatoeba Spanish dataset to the English (Project, 2022) and Sentiment140 dataset (Go et al., 2009) into Hinglish using Google translate. Further, to enhance the model's performance, they perform the validation tuning on the task-specific validation dataset and use a multi-run ensemble (Koehn, 2020) to combine multiple model's best checkpoints.
7. **UEDIN** (Kirefu et al., 2022): Their submission focuses on data generation using back-translation from monolingual resources. For *M2CM* subtask, they explore the impact of constrained and unconstrained decoding strategies. They use the Samanantar corpus (Ramesh et al., 2022) as an external resource for back-translation. For *CM2M* subtask, they explore several pretraining techniques, ranging from simple initialization from existing machine translation models to aligned augmentation (Pan et al., 2021) which is a denoising-based pretraining technique.

4 Results and Analysis

In this section, we present the results from automatic and human-based evaluation of the submissions from the seven teams. As discussed in Section 2.3, we use ROUGE-L F1 score (R-L) and Word Error Rate (WER) for automatic evaluation. R-L score can vary from 0 to 1 whereas WER can take a value greater than or equal to 0. A high R-L score and a low WER score are preferred.

Table 2 shows the results of the automatic evaluation for both subtasks. For *M2CM* subtask, the Gui team's submission achieves best R-L score whereas the team UEDIN and SIT-NMT achieve the second and third best R-L scores respectively. SIT-NMT's submission outperforms the other systems and scores the lowest WER for this subtask. For *CM2M* subtask, SIT-NMT is the best-performing team followed by UEDIN and MUCS on both metrics.

Table 3 shows the human-based evaluation of the submissions from different teams. The evaluation policy is given in Table 1. Following the evaluation policy, we evaluate the output of 20 samples for each subtask from each team. SIT-NMT ranks first

Team	*M2CM*		*CM2M*	
	R-L	WER	R-L	WER
Baseline	0.280	0.926	0.250	1.021
CNLP	0.238	0.926	0.330	0.88
DC	0.033	1.560	0.061	1.694
Gui	0.616	0.633	0.414	0.808
MUCS	0.358	0.760	0.550	0.647
NICT	0.462	0.792	0.528	0.715
SIT-NMT	0.57	0.547	0.629	0.607
UEDIN	0.579	0.561	0.621	0.624

Table 2: Evaluation results on the test set. We color code the best, second best, and third best team on a given metric for a subtask.

on both subtasks followed by UEDIN. Gui stood at the third position for *M2CM* task whereas MUCS is ranked third for *CM2M* subtask. Interestingly, MUCS and NICT get a consistent one score showing poor quality output consisting of lexical and grammatical mistakes. It further highlights the inefficacy of evaluation metrics for code-mixed natural language generation tasks as pointed out in several previous works (Garg et al., 2021; Srivastava and Singh, 2022).

Team	*M2CM*	*CM2M*
CNLP	2.1 ± 0.64	1.35 ± 0.74
DC	1.75 ± 0.71	1.55 ± 1.09
Gui	3.75 ± 1.20	1.8 ± 1.1
MUCS	1 ± 0	2.9 ± 1.51
NICT	1 ± 0	2.85 ± 1.30
SIT-NMT	3.85 ± 1.38	4.1 ± 1.07
UEDIN	3.85 + 1.53	3.75 + 1.16

Table 3: Human-based evaluation of submitted systems on the test set. We color code the best, second best, and third best team for a subtask.

Further, we analyze the submissions based on the dataset and the models used in the experiment. In Section 2.4, we have highlighted the two criteria for the submission to be considered as constrained. In Table 4, we summarize the submissions based on these two criteria.

We observe that almost all the teams have used at least one external dataset for both subtasks with Gui's submission for *CM2M* subtask being the only exception. We attribute this behavior to the fact we designed both subtasks in a low-resource setting. The submissions by four teams (i.e., CNLP, DC, NICT, and UEDIN) are completely unconstrained for both subtasks as they are using an external dataset and training their own system from scratch.

Team	*M2CM*		*CM2M*	
	OD	PAM	OD	PAM
CNLP	✗	✗	✗	✗
DC	✗	✗	✗	✗
Gui	✗	✓	✓	✓
MUCS	✗	✓	✗	✓
NICT	✗	✗	✗	✗
SIT-NMT	✗	✓	✗	✓
UEDIN	✗	✗	✗	✗

Table 4: Analysis of datasets and models used across submissions. Here, OD: organizer's dataset only and PAM: publicly available models.

5 Discussion

In this paper, we present the findings from the MixMT shared task. We hosted two subtasks involving a code-mixed language i.e. Hinglish. Given the low-resource nature of the code-mixed languages (and the subtasks), the majority of the submissions rely on data augmentation either synthetically or from other external sources. The lack of dedicated pre-trained models for code-mixed languages pushed the teams to explore the available alternatives along with bold attempts to train the models from scratch. We posit several open challenges with code-mixed machine translation such as creating large-scale parallel data, efficient data augmentation strategies, and robust evaluation measures. The insights and findings from this task will be useful to future works on machine translation involving code-mixed and low-resource languages. They will broaden the horizon for works on multilingual machine translation.

References

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code mixing: A challenge for language identification in the language of social media. In *Proceedings of the first workshop on computational approaches to code switching*, pages 13–23.

Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tammewar, Riyaz Ahmad Bhat, and Manish Shrivastava. 2015. Iiit-h system submission for fire2014 shared task on transliterated search. In *Proceedings of the Forum for Information Retrieval Evaluation*, FIRE '14, pages 48–53, New York, NY, USA. ACM.

Ondřej Bojar, Vojtěch Diatka, Pavel Rychlý, Pavel Straňák, Vít Suchomel, Aleš Tamchyna, and Daniel

Zeman. 2014. Hindencorp-hindi-english and hindi-only corpus for machine translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3550–3555.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Shardul Suryawanshi, Navya Jose, Elizabeth Sherly, and John P McCrae. 2020. Overview of the track on sentiment analysis for dravidian languages in code-mixed text. In *Forum for information retrieval evaluation*, pages 21–24.

Shuguang Chen, Gustavo Aguilar, Anirudh Srinivasan, Mona Diab, and Thamar Solorio. 2022. Calcs 2021 shared task: Machine translation for code-switched data. *arXiv preprint arXiv:2202.09625*.

Parth Chokhra. 2020. Hindi to hinglish corpus.

Raj Dabre. 2022. Nict at mixmt 2022: Synthetic code-mixed pre-training and multi-way fine-tuning for hinglish–english translation.

Raj Dabre and Eiichiro Sumita. 2021. Yanmtt: yet another neural machine translation toolkit. *arXiv preprint arXiv:2108.11126*.

Akshat Gahoi, Jayant Duneja, Anshul Padhi, Shivam Mangale, Saransh Rajput, Tanvi Kamble, Dipti Misra Sharma, and Vasudeva Varma. 2022. Gui at mixmt 2022 : English-hinglish: An mt approach for translation of code mixed data. *ArXiv*, abs/2210.12215.

Ayush Garg, Sammed Kagi, Vivek Srivastava, and Mayank Singh. 2021. Mipe: A metric independent pipeline for effective code-mixed nlg evaluation. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 123–132.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009.

Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2020. A semi-supervised approach to generate the code-mixed text using pre-trained encoder and transfer learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2267–2280.

Asha Hegde and Hosahalli Lakshmaiah Shashirekha. 2022. Mucs@mixmt: indictrans-based machine translation for hinglish text.

Abdul Rafae Khan, Hrishikesh Kanade, Girish Amar Budhrani, Preet Jhanglani, and Jia Xu. 2022. Sit at mixmt 2022: Fluent translation built on giant pre-trained models. *arXiv preprint arXiv:2210.11670*.

Faheem Kirefu, Vivek Iyer, Pinzhen Chen, and Laurie Burchell. 2022. The university of edinburgh's submission to the wmt22 code-mixing shared task (mixmt). *ArXiv*, abs/2210.11309.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Philipp Koehn. 2020. *Neural machine translation*. Cambridge University Press.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The iit bombay english-hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Sahinur Rahman Laskar, Rahul Singh, Shyambabu Pandey, Riyanka Manna, Partha Pakray, and Sivaji Bandyopadhyay. 2022. Cnlp-nits-pp at mixmt 2022: Hinglish–english code-mixed machine translation.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Paul Michel and Graham Neubig. 2018. Mtnt: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 543–553.

Carol Myers-Scotton. The matrix language frame model: Development and responses.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 244–258.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas Pykl, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. 2020. Semeval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. In *Proceedings of the fourteenth workshop on semantic evaluation*, pages 774–790.

Tatoeba Project. 2022. Tab-delimited bilingual sentence pairs.

Lekan Raheem and Maab Elrashid. 2022. Domain curricula for code-switched mt at mixmt 2022. *arXiv preprint arXiv:2210.17463*.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2022. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

Mohd Sanad Zaki Rizvi, Anirudh Srinivasan, Tanuja Ganu, Monojit Choudhury, and Sunayana Sitaram. 2021. Gcm: A toolkit for generating synthetic code-mixed text. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 205–211.

Brian Roark, Lawrence Wolf-Sonkin, Christo Kirov, Sabrina J Mielke, Cibu Johny, Isin Demirsahin, and Keith Hall. 2020. Processing south asian languages written in the latin script: the dakshina dataset. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2413–2423.

Vivek Srivastava and Mayank Singh. 2020. Phinc: A parallel hinglish social media code-mixed corpus for machine translation. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 41–49.

Vivek Srivastava and Mayank Singh. 2021a. Hinge: A dataset for generation and evaluation of code-mixed hinglish text. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 200–208.

Vivek Srivastava and Mayank Singh. 2021b. Quality evaluation of the low-resource synthetically generated code-mixed hinglish text. *INLG 2021*, page 314.

Vivek Srivastava and Mayank Singh. 2022. Code-mixed nlg: resources, metrics, and challenges. In *5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD)*, pages 328–332.

S Thara and Prabaharan Poornachandran. 2021. Transformer based language identification for malayalam-english code-mixed text. *IEEE Access*, 9:118837–118850.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2021. Automatic bilingual markup transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3524–3533, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018. A dataset for document grounded conversations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 708–713.

Findings of the Word-Level AutoCompletion Shared Task in WMT 2022[*]

Francisco Casacuberta [1] **George Foster** [2] **Guoping Huang** [3] **Philipp Koehn** [4,5]
Geza Kovacs [6] **Lemao Liu** [3] **Shuming Shi** [3] **Taro Watanabe** [7] **Chengqing Zong** [8]
[1] Universitat Politècnica de València [2] Google [3] Tencent AI Lab [4] Johns Hopkins University
[5] Meta AI [6] LILT [7] Nara Institute of Science and Technology
[8] Institute of Automation, Chinese Academy of Sciences

Abstract

Recent years have witnessed rapid advancements in machine translation, but the state-of-the-art machine translation system still can not satisfy the high requirements in some rigorous translation scenarios. Computer-aided translation (CAT) provides a promising solution to yield a high-quality translation with a guarantee. Unfortunately, due to the lack of popular benchmarks, the research on CAT is not well developed compared with machine translation. In this year, we hold a new shared task called Word-level AutoCompletion (WLAC) for CAT in WMT. Specifically, we introduce some resources to train a WLAC model, and particularly we collect data from CAT systems as a part of test data for this shared task. In addition, we employ both automatic and human evaluations to measure the performance of the submitted systems, and our final evaluation results reveal some findings for the WLAC task.

1 Introduction

In past decades, the machine translation community has witnessed a significant evolution from statistical machine translation (Koehn et al., 2003; Chiang, 2005; Koehn, 2009b) to neural machine translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2015; Wu et al., 2016; Gehring et al., 2017; Vaswani et al., 2017). NMT has achieved a rapid and tremendous advancement in translation performance (Barrault et al., 2019). Despite its success in many real-world applications, its translation quality still can not satisfy the high requirements in some scenarios. In such rigorous scenarios, one promising approach is to leverage machines to assist human translation, such as Computer-aided Translation (CAT) (Bowker, 2002; Koehn, 2009a; Foster et al., 1997; Langlais et al., 2000; Barrachina et al., 2009; Alabau et al., 2014; Knowles and Koehn, 2016; Santy et al., 2019).

However, the development in CAT is much slower than in machine translation. For example, there are hundreds of research papers on machine translation in natural language processing conferences each year, whereas only a few papers on CAT are published. One of the main reasons is that there are few popular benchmarks or shared tasks for CAT research, which enable researchers to make continuous progress in this area. Consequently, in WMT this year, we hold a new shared task, Word-level AutoCompletion (WLAC), to facilitate the research in CAT. Generally, WLAC aims to autocomplete a word when a human translator types a sequence of characters (Huang et al., 2015; Li et al., 2021). As a basic functionality in many CAT systems, WLAC is used to accelerate the editing process for human translators and it plays an important role in CAT.

In this paper, we describe the overview for the shared task of WLAC in WMT 2022, such as task description, datasets, participants and their evaluations. The shared task involves two language pairs, including Chinese-English and German-English and it contains four subtasks corresponding to all four directional pairs. For data preparation, since it is too costly to collect realistic data with a considerable scale to train WLAC models, we follow the standard practice to construct the training data from a bilingual corpus by simulation. Moreover, to make the testing stage resemble the realistic scenario in CAT, we collect some data from two CAT systems as a part of test data. In this shared task, we receive 27 submissions in total for all subtasks from five participants which are quickly summarized in this paper. To evaluate the submissions, we particularly conduct human evaluation in addition to automatic evaluation. After evaluation, we finally obtain some findings from the submission results, which we hope may inspire future advancements for the WLAC task.

[*] The authors are listed alphabetically.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 812 - 820
December 7-8, 2022 ©2022 Association for Computational Linguistics

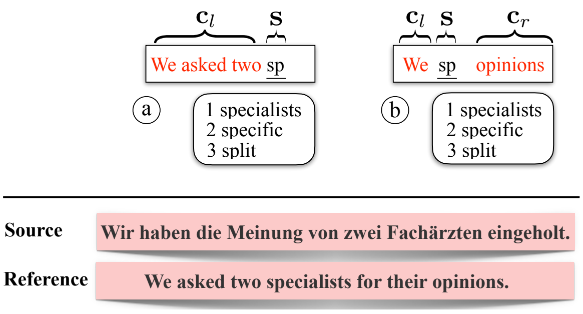


Figure 1: Illustration of WLAC task. The translation context $\boldsymbol{c}$ for a source sentence $\boldsymbol{x}$ includes the left context $\boldsymbol{c}_l$ and right context $\boldsymbol{c}_r$, underlined text "sp" is the human typed characters $\boldsymbol{s}$ and the words in the rounded rectangles are word-level autocompletion candidates.

2 Task Description and Data Preparation

2.1 Task Description

Suppose $\boldsymbol{x}$ is a source sequence, $\boldsymbol{s}$ is a sequence of human typed characters and $\boldsymbol{c} = (\boldsymbol{c}_l, \boldsymbol{c}_r)$ is a translation context. The translation pieces $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$ are on the left and right hand sides of $\boldsymbol{s}$, respectively. Formally, given the tuple $(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{s})$, the WLAC task aims to predict the target word w with $\boldsymbol{s}$ as its prefix, which is the most appropriate to be placed between $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$ (Huang et al., 2015; Li et al., 2021).

To make the task more general in real-world scenarios, WLAC task assumes that the left context $\boldsymbol{c}_l$ and right context $\boldsymbol{c}_r$ can be empty, which leads to the following four types of context:

- Zero-context: both $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$ are empty;
- Suffix: $\boldsymbol{c}_l$ is empty;
- Prefix: $\boldsymbol{c}_r$ is empty;
- Bi-context: neither $\boldsymbol{c}_l$ nor $\boldsymbol{c}_r$ is empty.

Figure 1 ⓐ and ⓑ show two examples about the WLAC task. According to the above criterion, Figure 1 ⓐ belongs to Prefix type and Figure 1 ⓑ belongs to Bi-context type.

2.2 Data Preparation

The WLAC task in WMT2022 involves following two language pairs: English⇔Chinese and English⇔German. Each language pair corresponds to two directional subtasks, leading to four subtasks.

Training Data In fact, it is too costly to manually annotate the training dataset consisting of tuples $\langle \boldsymbol{x}, \boldsymbol{c}, \boldsymbol{s}, w \rangle$ for WALC task. We alternatively follow Li et al. (2021) to construct the simulated training data for WLAC from existing bilingual data.[1] The key idea of such simulation is that it randomly samples a target word w and context $\boldsymbol{c}$ from the reference translation $\boldsymbol{y}$ of $\boldsymbol{x}$, a human typed sequence $\boldsymbol{c}$ for the target word w to obtain an example, e.g., a tuple of $\langle \boldsymbol{x}, \boldsymbol{c}, \boldsymbol{s}, w \rangle$.

	En-De	En-Zh
Sentences	4,465,840	15,886,041
Words	120M/114M	441M/395M

Table 1: The statistics of English-German and English-Chinese bilingual datasets for training.

For training on English-German language pair, we use the WMT14 En-De training dataset preprocessed by Stanford NLP Group[2], which consists of about 4.5M sentence pairs. For training on English-Chinese language pair, we take the "UN Parallel Corpus V1.0" dataset[3] from WMT17 consisting of 15M sentence pairs. We use Moses scripts[4] to tokenize English and German sentences and jieba[5] to segment Chinese words for each sentence. The detailed statistics of bilingual datasets are shown in Table 1.

Note that in this shared task, participants must use the above bilingual data and it is illegal to any other bilingual data beyond. However, to achieve better performance, any monolingual data is allowed as well as the pre-trained language models such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019).

Test Data To ensure authenticity and reliability, the test data for WLAC is not from existing open-source bilingual data. We create the test data by ourselves: the test datasets are obtained in two different ways, leading to two types of test data. One type (**Type I**) is the simulation on bilingual data similar to the creation of training data and the other type (**Type II**) is from CAT translation systems.

For the Type I test data, we first create a new bilingual dataset and then obtain the simulated tuples $\langle \boldsymbol{x}, \boldsymbol{c}, \boldsymbol{s}, w \rangle$ from the bilingual dataset. Specifically, to ensure that the ground-truth word w is not

[1] The scripts for simulation is available at https://github.com/lemaoliu/WLAC.

[2] https://nlp.stanford.edu/projects/nmt/data

[3] https://conferences.unite.un.org/UNCorpus/Home/DownloadOverview

[4] https://github.com/moses-smt/mosesdecoder

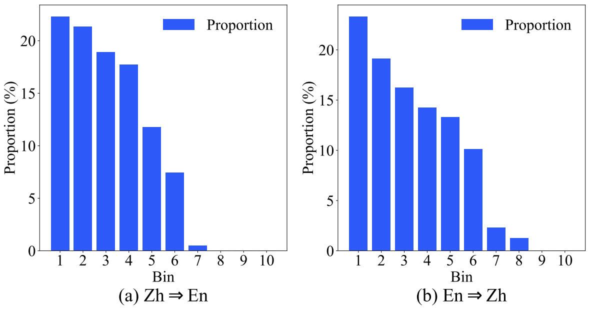
[5] https://github.com/fxsjy/jieba

	Zh⇒En	En⇒Zh	De⇒En	En⇒De
		Sentences		
Type I	5434	6122	5700	-
Type II	2109	1953	1996	13418
Overall	**7543**	**8075**	**7696**	**13418**
		Words		
Type I	615K/115K	662K/109K	519K/96K	-
Type II	242K/45K	237K/38K	203K/38K	437K/85K
Overall	**857K/161K**	**899K/147K**	**722K/134K**	**437K/85K**

Table 2: The statistics (number of sentences and words) of Zh⇒En, En⇒Zh, De⇒En and En⇒De test datasets. A/B denotes that A is the total number of source words in the source sentences and B is the total number of target words in the context.

	Zh⇒En	En⇒Zh	De⇒En	En⇒De
		Bi-context		
Type I	5102	5137	5313	-
Type II	2092	1676	1950	6514
Overall	**7194**	**6813**	**7263**	**6514**
		Prefix		
Type I	5330	5249	5686	-
Type II	2087	1645	1968	6319
Overall	**7417**	**6894**	**7654**	**6319**
		Suffix		
Type I	5053	5156	5382	-
Type II	2089	1674	1994	6571
Overall	**7142**	**6830**	**7376**	**6571**
		Zero-context		
Type I	5200	5137	5256	-
Type II	2098	1622	2047	6491
Overall	**7298**	**6759**	**7303**	**6491**

Table 3: The statistics (number of $\langle \boldsymbol{x}, \boldsymbol{c}, \boldsymbol{s}, w \rangle$ tuples) of different context types on WLAC test datasets (including both Type I and Type II parts).



(a) The proportion on Zh⇒En **(b)** The proportion on En⇒Zh

Figure 2: The proportion of the bins of w typed by human translators from CAT systems according to word frequency in bilingual corpus on German-English language pair. Bin 1 and Bin 10 respectively denote the most infrequent word bin and the most frequent bin.

exposed to the training data, we first crawl bilingual news from Internet websites in the latest 3 years. After crawling the raw bilingual data, we employ professional translators to check and screen the low quality bilingual data to obtain high-quality bilingual sentences. Finally, we follow the simulation way to obtain the training tuples $\langle \boldsymbol{x}, \boldsymbol{c}, \boldsymbol{s}, w \rangle$ based on the crawled bilingual data described above.

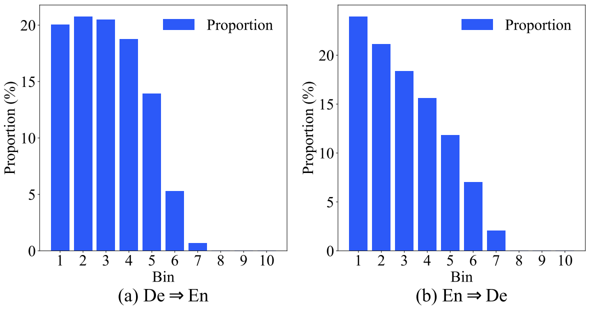
The Type II test data is collected from two CAT systems LILT[6] and TranSmart[7] (Huang et al., 2021). Specifically, given a source sentence $\boldsymbol{x}$, a human translator works on a CAT system to generate a translation $\boldsymbol{y}$. In the log file from the CAT system, only the information about $w \in \boldsymbol{y}$ typed by human is stored, while other dynamic information such as typed characters and context for each w is not available. Therefore, we create both $\boldsymbol{c}$ and $\boldsymbol{s}$ from $\boldsymbol{y}$ for each w by simulation as before. In other words, for each example $\langle \boldsymbol{x}, \boldsymbol{c}, \boldsymbol{s}, w \rangle$, both $\boldsymbol{c}$ and $\boldsymbol{s}$ are simulated but w is realistic. Note that each sentence from the Type II data is also not included in the training data.

For En⇒De task, the entire test data is the type II from the CAT system LILT. For Zh⇒En, En⇒Zh and De⇒En tasks, the test data is the combination of both types, i.e., some test data is Type I from the simulation over bilingual data and the other test data is Type II from the CAT system TranSmart.

To pre-process the test data (e.g., word tokenization), we adopt the same pre-processing way as used in training data. Table 2 summarizes the detailed statistics in terms of sentences and words for the test data, and Table 3 reports the number of ex-

[6] https://lilt.com

[7] https://transmart.qq.com



(a) The proportion on De⇒En **(b)** The proportion on En⇒De

Figure 3: The proportion of the bins of w typed by human translators from CAT systems according to word frequency in bilingual corpus on German-English language pair. Bin 1 and Bin 10 respectively denote the most infrequent word bin and the most frequent bin.

amples for four different context types in test data. Note that each source sentence $\boldsymbol{x}$ may correspond to multiple examples $\langle \boldsymbol{x}, \boldsymbol{c}, \boldsymbol{s}, w \rangle$ and thus, the total number of sentences in Table 2 is not the same as the total number of examples in Table 3.

Furthermore, one may be curious about the characteristics of the words typed by human translators. We understand the human typed words from the perspective of word frequency. We first group the target vocabulary into ten bins with equal size according to word frequency computed in the bilingual corpus, we collect all typed words w together and then assign a bin for each word, and finally we calculate the proportion of each bin. The statistics are shown in Table 2 and Table 3, where bin 1 denotes the most infrequent words while bin 10 denotes the most frequent words. From these tables, it is observed that human translators usually type infrequent words. This observation is reasonable because it is easy for machine translation systems to make a correct translation decision on a frequent word.

3 Evaluation Metric

We use both automatic evaluation and human evaluation to measure all submitted systems.

Automatic Evaluation To measure the performance of the submitted systems, we choose accuracy as the automatic evaluation metric (Li et al., 2021) as follows :

$$\text{ACC} = \frac{N_{match}}{N_{all}} \tag{1}$$

where N_{match} is the number of correct predicted words and N_{all} is the number of **all** test examples.

Although automatic evaluation is convenient, it still has some limitations because there may be multiple ground-truth words w (i.e., ground truth is a word set) which suffice to the constraint of $\boldsymbol{s}$ and are compatible with $\langle \boldsymbol{x}, \boldsymbol{c} \rangle$, especially for a short $\boldsymbol{c}$ and $\boldsymbol{s}$. For instance, when $\boldsymbol{c}$ and $\boldsymbol{s}$ are empty, any translation of a source word in $\boldsymbol{x}$ may be a ground-truth word if it suffices to the constraint of $\boldsymbol{s}$. Therefore, we additionally conduct human evaluation for more faithful evaluation on the submissions.

Human Evaluation Human evaluation is appealing, but it is too costly to evaluate all testing examples. Instead, we conduct human evaluation on a small subset of test data for efficiency. Specifically, for all four subtasks, we randomly sample **400** test examples derived from the Type II part of the test data as the human evaluation dataset. After participants submit their systems, we gather their predicted words to constitute a prediction set for each test example. Then we hire professional translators to annotate the correct ones in the prediction set. Finally, we use the manually annotated ground-truth word set to re-evaluate submitted systems and the human score is defined by the percentage of predicted words annotated as correct words by human. Since more than one target word can be annotated as the correct word, the human evaluation score is higher than the automatic score in general.

4 Submitted Systems and Results

In this year, there are five teams participating in this shared task and we receive 27 submissions from them. In this section, first, we quickly describe the participants and their submitted systems, then we present their evaluation results in terms of both automatic and human evaluations, and finally, we shed light on some findings according to evaluation results.

4.1 Participants and Submitted Systems

HW-TSC (Yang et al., 2022b) The Huawei Translation Services Center (HW-TSC) participates in Zh⇒En, De⇒En and En⇒De language directions. They model the WLAC task as a structured prediction (or generation) task, which iteratively generates a subword to compose the prediction word. Specifically, they first train a vanilla Transformer on machine translation task as a baseline. Then they fine-tune the baseline with WLAC data and BERT-style MLM data to get the final model.

Systems	Fullset	Subset	
	Acc. (Rank)	Human. (Rank)	Acc. (Rank)
HW-TSC	59.40 (#1)	91.25 (#1)	69.25 (#1)
THU IIGroup-1	54.05 (#2)	85.00 (#6)	59.75 (#6)
THU IIGroup-2	51.11 (#3)	83.75 (#7)	57.25 (#7)
DCU-NCI-4	50.41 (#4)	86.75 (#3)	63.25 (#2)
DCU-NCI-3	50.26 (#5)	86.75 (#3)	62.25 (#3)
DCU-NCI-2	49.35 (#6)	86.00 (#5)	61.75 (#5)
DCU-NCI-1	49.06 (#7)	87.00 (#2)	62.00 (#4)

Table 4: Official results of WLAC task for Zh⇒En. Acc. and Human. represent automatic and human evaluations, respectively. Rank denotes the ranking according to the corresponding metric.

Systems	Fullset	Subset	
	Acc. (Rank)	Human. (Rank)	Acc. (Rank)
THU IIGroup-1	53.98 (#1)	83.25 (#1)	54.50 (#1)
THU IIGroup-2	48.90 (#2)	77.50 (#2)	48.75 (#2)
DCU-NCI-2	31.94 (#3)	57.75 (#3)	37.75 (#4)
DCU-NCI-1	31.94 (#4)	57.25 (#4)	38.00 (#3)

Table 5: Official results of WLAC task for En⇒Zh. Acc. and Human. represent automatic and human evaluations, respectively. Rank denotes the ranking according to the corresponding metric.

It is worth noting that they use a character embedding method to encode the information of a human typed sequence to the model. Moreover, they adopt some basic strategies to improve the performance, including back translation, averaging and ensemble techniques.

PRHLT (Ángel Navarro et al., 2022) The team of PRHLT submits their systems for De⇒En and En⇒De subtasks. They first cast the WLAC task as a segment-based IMT task. More concretely, they consider the translation context as the sequence of segments validated by the user in IMT and the sequence of human typed characters as partially-typed word correction. They experiment with both RNN architecture and Transformer architecture.

DCU-NCI (Moslem et al., 2022) DCU-NCI proposes to address the WLAC task with the help of pre-trained NMT models and available libraries, which is a new way to solve the WLAC task. Their systems do not need any additional training to address the WLAC task.Specifically, they use OPUS pre-trained models[8] and employ CTranslate2 [9] as an inference engine. During the decoding stage, they find that random sampling restricted with the best 10 candidates perform better than beam search. Furthermore, they also try to adopt different sampling temperatures (ST) to change the randomness of the generation. We denote the system trained with ST=1.0 as DCU-NCI-1, the system with ST=1.3 as DCU-NCI-2, the system with ST=1.3 and detokenization as DCU-NCI-3 and the system trained with ST=1.0 and detokenization as DCU-NCI-4.

Lingua Custodia (Ailem et al., 2022) The team of Lingua Custodia submits systems for De⇒En and En⇒De tracks. They also treat the WLAC task as a structured prediction task and adopt the Transformer architecture for generation. Specifically, they use a Transformer Encoder to encode the source sentence, translation context and human typed characters, and a Transformer Decoder to generate a sequence of subwords to constitute a target word step by step. In addition, they propose several data-cleaning strategies to pre-process the bilingual translation data. We denote the system trained with the initial corpus as Lingua Custodia-1 and the system trained with the cleaned corpus as Lingua Custodia-2.

[8] https://github.com/Helsinki-NLP/Tatoeba-Challenge

[9] https://github.com/OpenNMT/CTranslate2

Systems	Fullset	Subset	
	Acc. (Rank)	Human. (Rank)	Acc. (Rank)
HW-TSC	62.06 (#1)	87.50 (#3)	78.00 (#3)
DCU-NCI-1	61.44 (#2)	88.50 (#2)	80.50 (#1)
DCU-NCI-2	60.92 (#3)	88.75 (#1)	79.00 (#2)
Lingua Custodia-1	57.36 (#4)	76.75 (#5)	67.50 (#5)
THU IIGroup-1	57.27 (#5)	78.75 (#4)	69.75 (#4)
Lingua Custodia-2	54.85 (#6)	74.50 (#7)	63.50 (#7)
THU IIGroup-2	54.32 (#7)	76.25 (#6)	66.50 (#6)
PRHLT	39.02 (#8)	51.25 (#8)	44.25 (#8)

Table 6: Official results of WLAC task for De⇒En. Acc. and Human. represent automatic and human evaluations, respectively. Rank denotes the ranking according to the corresponding metric.

Systems	Fullset	Subset	
	Acc. (Rank)	Human. (Rank)	Acc. (Rank)
HW-TSC	63.82 (#1)	79.00 (#1)	66.75 (#1)
DCU-NCI-1	58.94 (#2)	67.25 (#2)	56.00 (#2)
DCU-NCI-2	58.49 (#3)	65.50 (#3)	56.75 (#3)
Lingua Custodia-1	48.97 (#4)	61.75 (#4)	52.25 (#4)
Lingua Custodia-2	48.44 (#5)	61.00 (#5)	50.75 (#5)
THU IIGroup-1	41.83 (#6)	55.50 (#6)	46.00 (#6)
THU IIGroup-2	40.69 (#7)	53.50 (#7)	44.75 (#7)
PRHLT	33.97 (#8)	45.75 (#8)	37.00 (#8)

Table 7: Official results of WLAC task for En⇒De. Acc. and Human. represent automatic and human evaluations, respectively. Rank denotes the ranking according to the corresponding metric.

THU IIGroup (Yang et al., 2022a) THU IIGroup participates in Zh⇒En, En⇒Zh, De⇒En and En⇒De directions. They propose a generator-reranker framework to tackle the WLAC task. Specifically, they adopt the baseline model based on Transformer as a generator to yield a set of candidate words. Moreover, they additionally train a reranking model to rerank the candidate words to get the final prediction. We denote the generator as THU IIGroup-1 and the reranker as THU IIGroup-2.

Summary on submitted systems All submitted systems in this year choose the powerful Transformer architecture by stacking multiple layers of attention as the backbone for the WLAC task. To tackle the constraint of the human typed character sequence, some submitted systems consider it as a hard constraint while others (HW-TSC and Lingua Custodia) considering it as a soft constraint: they differ in that the model architecture in the former is aware of the constraints but the later matters. In addition, most systems formalize the WLAC task as a classification task where the target word w is actually a label, but one system (HW-TSC) treats WLAC as a structured prediction task: the target w is decomposed into a sequence of BPE units and it is beneficial to predict the out-of-vocabulary words.

4.2 Evaluation Results

Since human evaluation is only conducted on the partial test dataset consisting of 400 examples and automatic evaluation can be evaluated on both the full and partial test datasets, we evaluate all the submitted systems on two different types of test data, i.e., full test data set and partial test data set as follows. All of their results on Zh⇒En, En⇒Zh, De⇒En and En⇒De are listed in Table 4,5,6 and 7.

Results on Full Test Set From the four tables, it can be shown that the systems of HW-TSC shows impressive performance and achieve the best for Zh⇒En, De⇒En and En⇒De, and THU IIGroup yields the best performance for En⇒Zh. As we can see, there are some gaps in performance among different systems, which means there is a significant opportunity for growth in the WLAC task.

Results on Partial Test Set As described in Section 3, it is not surprising that human evaluation scores are much higher than automatic evaluation scores. In addition, it is observed that on the partial test set, the human evaluation results are almost in line with the automatic evaluation result although there indeed is a slight inconsistency. This fact demonstrates that automatic evaluation metric can act as a good alternative for evaluation. Moreover, it is interesting that, in terms of automatic evaluation, the rankings between the full and partial test datasets are clearly different on Zh⇒En, although they are mostly consistent on other tasks. This observation indicates that a small test dataset may lead to a biased conclusion.

4.3 Discussion

In this section, we shed light on some key findings among all the submitted systems which we hope will push forward the development of the WLAC task in the future.

First, it would be preferable to treat the WLAC task as a structured prediction task rather than a classification task according to the prediction accuracy. One advantage of the structured prediction perspective is that it can decompose the predicted word into a sequence of tokens at the subword level to tackle out-of-vocabulary words. This is appealing specially because most of the typed words by human translators are low frequent words as observed in our analysis. However, it is noteworthy that a structured predition model requires more computing time than a classification model during the inference stage.

Second, WLAC task may benefit from NMT based pre-training. It is noticed that one participant employs such a pre-training strategy: it first trains a standard NMT model on the bilingual dataset and then it fine-tunes the model with the WLAC data to obtain a WLAC model. It is reasonable since in NMT task, every token in the target-side serves as a label, while in WLAC task, only the target token serves as a label. The former can facilitate the training procedure and provide a good weight initialization for WLAC tailored model.

Third, leveraging monolingual data is a common practice to improve the performance in many NLP tasks, including machine translation. For example, a pre-trained model trained on monolingual data such as XLM (Lample and Conneau, 2019) and MASS (Song et al., 2019) are successful to improve translation quality, and back translation (Sennrich et al., 2015; Edunov et al., 2018) is also an effective strategy by construction synthetic bilingual data from target monolingual data. In WLAC task, one participant tries to enhance the WLAC model by using back translation similar to NMT and it is promising to design new ways customized for WLAC.

5 Conclusion

Word-level AutoCompletion is a basic functionality in computer-aided translation systems to facilitate the editing efficiency for translators. In WMT this year, the Word-level AutoCompletion shared task is introduced and it covers two language pairs including four directional subtasks. We provide high-quality test datasets and human evaluation to evaluate different systems fairly. On all subtasks we receive 27 submissions from five participants which address the WLAC task from different perspectives. Automatic and human evaluations on these submissions reveal some key findings which may provide valuable insights for future research on this task. Finally, we hope that WLAC task will attract more researchers to participate in the exploration of computer-aided translation.

Acknowledgements

We would like to thank Huayang Li for valuable discussions and Cheng Yang for data pre-processing on the shared task, and especially appreciate the annotators for their creating test data and manual evaluation on the submitted systems. In addition, we thank the participants for their contributions on this shared task.

References

Melissa Ailem, Jinghsu Liu, Jean-Gabriel Barthélemy, and Raheel Qader. 2022. Lingua custodia's participation at the wmt 2022 word-level auto-completion shared task. In *Proceedings of the Seventh Conference on Machine Translation Shared Tasks Papers*. Association for Computational Linguistics.

Vicent Alabau, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Ulrich Germann, Jesús González-Rubio, Robin Hill, Philipp Koehn, Luis A Leiva, et al. 2014. Casmacat: A computer-assisted translation workbench. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio L. Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and Juan Miguel Vilar. 2009. Statistical approaches to computer-assisted translation. *Comput. Linguistics*, 35(1):3–28.

Loïc Barrault, Ondřej Bojar, Marta R Costa-Jussa, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, et al. 2019. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61.

Lynne Bowker. 2002. *Computer-aided translation technology: A practical introduction*. University of Ottawa Press.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 263–270.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 489–500. Association for Computational Linguistics.

George Foster, Pierre Isabelle, and Pierre Plamondon. 1997. Target-text mediated interactive machine translation. *Machine Translation*, 12(1):175–194.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1243–1252.

Guoping Huang, Lemao Liu, Xing Wang, Longyue Wang, Huayang Li, Zhaopeng Tu, Chengyan Huang, and Shuming Shi. 2021. Transmart: A practical interactive machine translation system. *CoRR*, abs/2105.13072.

Guoping Huang, Jiajun Zhang, Yu Zhou, and Chengqing Zong. 2015. A new input method for human translators: integrating machine translation effectively and imperceptibly. In *IJCAI*.

Rebecca Knowles and Philipp Koehn. 2016. Neural interactive translation prediction. In *12th Conferences of the Association for Machine Translation in the Americas: MT Researchers' Track, AMTA 2016, Austin, TX, USA, October 28 - November 1, 2016*, pages 107–120. The Association for Machine Translation in the Americas.

Philipp Koehn. 2009a. A process study of computer-aided translation. *Machine Translation*, 23(4):241–263.

Philipp Koehn. 2009b. *Statistical machine translation*. Cambridge University Press.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Philippe Langlais, George Foster, and Guy Lapalme. 2000. Transtype: a computer-aided translation typing system. In *ANLP-NAACL 2000 Workshop: Embedded Machine Translation Systems*.

Huayang Li, Lemao Liu, Guoping Huang, and Shuming Shi. 2021. GWLAN: general word-level autocompletion for computer-aided translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4792–4802. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yasmin Moslem, Rejwanul Haque, and Andy Way. 2022. Word-level auto-completion: What can we achieve out of the box? In *Proceedings of the Seventh Conference on Machine Translation Shared Tasks Papers*. Association for Computational Linguistics.

Ángel Navarro, Miguel Domingo, and Francisco Casacuberta. 2022. Prhlt's submission to wlac 2022. In *Proceedings of the Seventh Conference on Machine Translation Shared Tasks Papers*. Association for Computational Linguistics.

Sebastin Santy, Sandipan Dandapat, Monojit Choudhury, and Kalika Bali. 2019. INMT: interactive neural machine translation prediction. In *Proceedings* *of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019 - System Demonstrations*, pages 103–108. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Cheng Yang, Siheng Li, Chufan Shi, and Yujiu Yang. 2022a. Iigroup submissions for wmt22 word-level autocompletion task. In *Proceedings of the Seventh Conference on Machine Translation Shared Tasks Papers*. Association for Computational Linguistics.

Hao Yang, Hengchao Shang, Zongyao Li, Daimeng Wei, Xianghui He, Xiaoyu Chen, Zhengzhe Yu, Jiaxin Guo, Jinlong Yang, Shaojun Li, Yuanchang Luo, Yuhao Xie, Lizhi Lei, and Ying Qin. 2022b. Hw-tsc's submissions to the wmt22 word-level auto completion task. In *Proceedings of the Seventh Conference on Machine Translation Shared Tasks Papers*. Association for Computational Linguistics.

Findings of the WMT 2022 Shared Task on Translation Suggestion

Zhen Yang, Fandong Meng, Yingxue Zhang, Ernan Li, and Jie Zhou
Pattern Recognition Center, WeChat AI, Tencent Inc, Beijing, China
{zieenyang, fandongmeng, yxuezhang, cardli, withtomzhou}@tencent.com

Abstract

We report the result of the first edition of the WMT shared task on Translation Suggestion (TS). The task aims to provide alternatives for specific words or phrases given the entire documents generated by machine translation (MT). It consists two sub-tasks, namely, the naive translation suggestion and translation suggestion with hints. The main difference is that some hints are provided in sub-task two, therefore, it is easier for the model to generate more accurate suggestions. For sub-task one, we provide the corpus for the language pairs English-German and English-Chinese. And only English-Chinese corpus is provided for the sub-task two.

We received 92 submissions from 5 participating teams in sub-task one and 6 submissions for the sub-task 2, most of them covering all of the translation directions. We used the automatic metric BLEU for evaluating the performance of each submission.

1 Introduction

Computer-aided translation (CAT) (Barrachina et al., 2009; Green et al., 2014; Knowles and Koehn, 2016; Santy et al., 2019) has attained more and more attention for its promising ability in combining the high efficiency of machine translation (MT) (Cho et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017) and high accuracy of human translation (HT). A typical way for CAT tools to combine MT and HT is PE (Green et al., 2013; Zouhar et al., 2021), where the human translators are asked to provide alternatives for the incorrect word spans in the results generated by MT. To further reduce the post-editing time, researchers propose to apply TS into PE, where TS provides the sub-segment suggestions for the annotated incorrect word spans in the results of MT, and their extensive experiments show that TS can substantially reduce translators' cognitive loads and the post-editing time (Wang et al., 2020; Lee et al., 2021).

As there is no explicit and formal definition for TS, we observe that some previous works similar or related to TS have been proposed (Alabau et al., 2014; Santy et al., 2019; Wang et al., 2020; Lee et al., 2021). However, there are two main pitfalls for these works in this line. First, most conventional works only focus on the overall performance of PE but ignore the exact performance of TS. This is mainly because the golden corpus for TS is relatively hard to collect. As TS is an important sub-module in PE, paying more attention to the exact performance of TS can boost the performance and interpretability of PE. Second, almost all of the previous works conduct experiments on their in-house datasets or the noisy datasets built automatically, which makes their experiments hard to be followed and compared. Additionally, experimental results on the noisy datasets may not truly reflect the model's ability on generating the right predictions, making the research deviate from the correct direction. Therefore, the community is in dire need of a benchmark for TS to enhance the research in this area. To address the limitations mentioned above and spur the research in TS, we make our efforts to construct a high-quality benchmark dataset with human annotation, named *WeTS*,[1] which covers four different translation directions.

The main motivation of this shared task is two-fold. The first goal is to analyze the challenges in the area of TS, which can provide some new directions for the further researches and applications in this area. Secondly, we want to make the researchers notice the gaps between the golden and automatically generated synthetic corpus. And we want to see the performance of different techniques on the golden corpus. As the source and translation sentence are both the inputs of TS, it is interesting to see how the interactions between the source and

[1] *WeTS*: We Establish a benchmark for Translation Suggestion

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 821 - 829
December 7-8, 2022 ©2022 Association for Computational Linguistics

translation sentences can improve the final suggestions.

In order to evaluate the quality of the participating systems, we use the automatic metric, BLEU (Papineni et al., 2002). Specifically, we adopt the widely used toolkit, sacrebleu (Post, 2018) to calculate the BLEU score for the top-1 suggestion against the reference sentences.[2] For Chinese, the BLEU score is calculated on teh character with the default tokenizer for Chinese. As for English, the BLEU score is calcualted on the case-sensitive words with the default tokenizer 13a.

Five teams participated in this first campaign of the Translation Suggestion shred task, most of them cover the four translation directions. We will describe each system which submits the technical paper in detail.

2 Task Description

This section describes the task definition in the first edition of TS shared task. We finely divide the task of TS into two sub-tasks, namely *vanilla TS* and *TS with hints*, according to whether the translators' hints are considered.

Vanilla TS. Given the source sentence $\boldsymbol{x} = (x_1, \ldots, x_s)$, the translation sentence $\boldsymbol{m} = (m_1, \ldots, m_t)$, the incorrect words or phrases $\boldsymbol{w} = \boldsymbol{m}_{i:j}$ where $1 \leq i \leq j \leq t$, and the correct alternative $\boldsymbol{y}$ for $\boldsymbol{w}$, the task of *vanilla TS* is optimized to maximize the conditional probability of $\boldsymbol{y}$ as follows:

$$P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{m}^{-\boldsymbol{w}}, \theta) \tag{1}$$

where θ represents the model parameter, and $\boldsymbol{m}^{-\boldsymbol{w}}$ is the masked translation where the incorrect word span $\boldsymbol{w}$ is replaced with a placeholder. [3]

TS with Hints. In the sub-task *TS with hints*, the hints of translators are considered as some soft constraints for the model, and the model is expected to generate suggestions meeting these constraints. The format of the translator's hint is very flexible, which usually requires only a few types on the keyboard by the translator. For English and German, the hints can be the character sequence which includes the initials of words in the correct alternative. As for Chinese, the hints can be the character sequence which includes the initials of the phonetics of words in the correct alternative. In this setting, the model is optimized as:

$$P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{m}^{-\boldsymbol{w}}, \boldsymbol{h}, \theta) \tag{2}$$

where $\boldsymbol{h}$ indicates the hints provided by translators.

Related tasks. Some similar techniques have been explored in CAT. Green et al. (2014) and Knowles and Koehn (2016) study the task of so-called translation prediction, which provides predictions of the next word (or phrase) given a prefix. Huang et al. (2015) and Santy et al. (2019) further consider the hints of the translator in the task of translation prediction. Compared to TS, the most significant difference is the strict assumption of the translation context, i.e., the prefix context, which severely impedes the use of their methods under the scenarios of PE. Lexically constrained decoding which completes a translation based on some unordered words, relaxes the constraints provided by human translators from prefixes to general forms (Hokamp and Liu, 2017; Post and Vilar, 2018; Kajiwara, 2019; Susanto et al., 2020). Although it does not need to re-train the model, its low efficiency makes it only applicable in scenarios where only a few constraints need to be applied. Recently, Li et al. (2021) study the problem of auto-completion with different context types. However, they only focus on the word-level auto-completion, and their experiments are also conducted on the automatically constructed datasets.

3 Data Description

This section introduces the proposed dataset *WeTS* used in the shred task, which is a golden corpus for four translation directions, including English-to-German, German-to-English, Chinese-to-English and English-to-Chinese.

Translation Direction	Train	Valid	Test
En⇒De	14,957	1000	1000
De⇒En	11,777	1000	1000
Zh⇒En	21,213	1000	1000
En⇒Zh	15,769	1000	1000

Table 1: The sizes for cases in train/valid/test sets. "En⇒De" refers to the direction of English-to-German, and "En⇒Zh" refers to English-to-Chinese.

[2] https://github.com/mjpost/sacrebleu

[3] $\boldsymbol{w}$ is null if i equals j, and the model will predict whether some words need to be inserted in position i.

Source Sentence	他们也许并不知道这是一个“假理财”骗局，但也察觉到了诸多可疑之 ta men ye xu bing bu zhi dao zhe shi yi ge jia li cai pian ju, dan ye cha jue dao le zhu duo ke yi zhi chu 处，然而最终还是按照张颖的指使进行了违法违规操作。 ran er zui zhong hai shi an zhao zhang ying de zhi shi jin xing le wei fa wei gui cao zuo
Translation	They may not know this is a "fake financial management" scam, but also aware of many suspicious, and ultimately conduct illegal operations according to Zhang Ying's instructions.
Suggestions	1. suspects 2. doubtful points 3. questionable points

Figure 1: One training example in *WeTS*. For the incorrect word "suspicious" (in red color), there are three correct suggestions. For readability, we also provide the Chinese pinyin format for the Chinese sentence (in blue color).

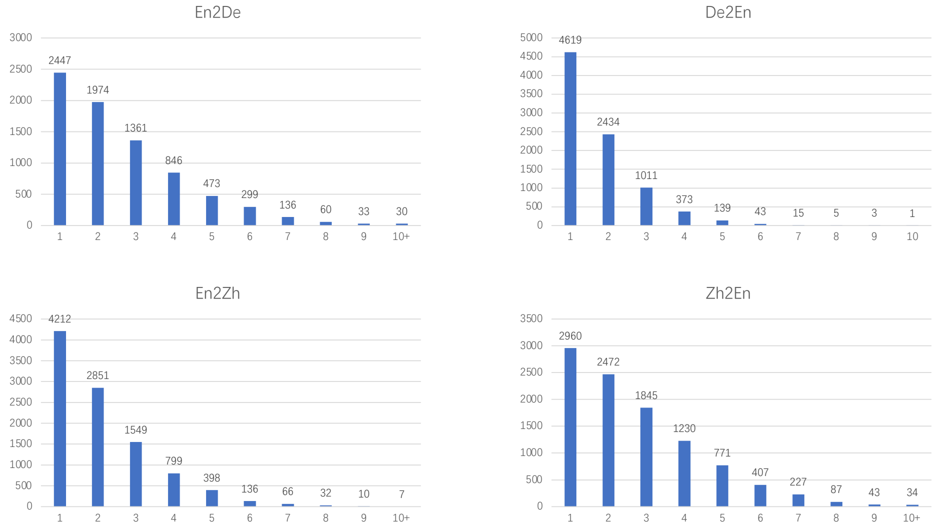


Figure 2: The number of incorrect span in each annotated example.

3.1 Annotation Guidelines

It is non-trivial for annotators to locate the incorrect word spans in the MT sentence. The main difficulty is that, the concept of “translation error" is ambiguous and each translator has his own understanding about translation errors. To easier the annotation workload and reduce the possibility of making errors, we group the translation errors on which we aim to focus into three macro categories:

- Under-translation or over-translation: While the problem of under-translation or over-translation has been alleviated with the popularity of Transformer, it is still one of the main mistakes in NMT and seriously destroys the readability of the translation.

- Semantic errors: For the semantic error, we mean that some source words are incorrectly translated according to the semantic context, such as the incorrect translations for entities, proper nouns, and ambiguous words. Another branch of semantic mistake is that the source words or phrases are only translated superficially and the semantics behind are not translated well.

- Grammatical or syntactic errors: Such errors usually appear in translations of long sentences, including the improper use of tenses, passive voice, syntactic structures, etc.

Another key rule for translators is that annotating the incorrect span as local as possible, as generating correct alternatives for long sequences is much harder than that of shorter sequences.

3.2 Data Construction

As the starting point, we collect the monolingual corpora for English and German from the raw Wikipedia dumps, and extract Chinese monolingual corpus from various online news publications. We first clean the monolingual corpora with a language detector to remove sentences belonging to

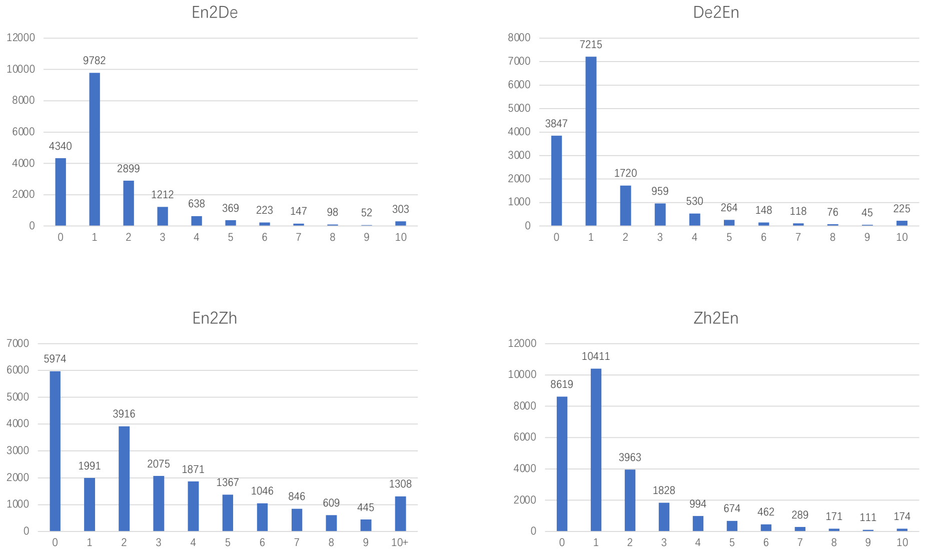


Figure 3: The length of the incorrect span.

other languages.[4] For all monolingual corpora, we remove sentences that are shorter than 20 words or longer than 80 words. In addition, sentences which exist in the available parallel corpora are also removed. Then, we get the translations by feeding the cleaned monolingual corpus into the corresponding fully-trained NMT model. The NMT models for English-German language pairs are trained on the parallel corpus of WMT14 English-German. For Chinese-English directions, the NMT models are trained with the combination between the WMT19 English-Chinese[5] and the same amount of in-house corpus. [6]

Finally, the translators are required to mark the incorrect word spans in the translation sentence and provide at least one alternative for each incorrect span, by using the annotation guidelines. The team is composed by eight annotators with high expertise in translation and each example has been assigned to three experts. There are two phases of agreement computations. In the first phase, an annotation is considered in agreement among the experts if and only if they capture the same incorrect word spans. If one annotation passes the first agreement computation, it will be assigned to other three experts in charge of selecting the right alternatives from the previous annotation. In the second phase of agreement computation, an annotation is considered in agreement among the experts if and only if they select the same right alternatives. With the two-phase agreement checking, we ensure the high quality of the annotated examples. For the annotated examples with multiple incorrect word spans, we can extract multiple examples which have the same source and translation sentences, but different incorrect word span and the corresponding suggestions. Finally the extracted examples are randomly shuffled and then split into the training, validation and test sets.[7] One training example for the translation direction of Chinese-to-English is presented in Figure 1 and the sizes for the train/valid/test sets in *WeTS* are collected in Table 1.

3.3 Detailed Statistics

The number of the incorrect span Each annotated example may contain multiple incorrect spans, we show the number of the incorrect span in each annotated example as Figure 2. We can see that most examples have only a few incorrect spans, and there are more than 70 percent examples containing less than 3 incorrect spans for each translation direction.

The length of the incorrect span Figure 3 represents the length distribution of the incorrect spans. We can find that most of the incorrect spans contain less than 3 words or Chinese characters. This is mainly because of our key rule for annotating the incorrect span as local as possible. Additionally, for all of the four translation directions, the

[4] `https://github.com/Mimino666/langdetect`

[5] `https://www.statmt.org/wmt19/translation-task.html`

[6] We have released the models and inference scripts utilized here to make our results easy reproduced.

[7] To keep the fairness of *WeTS*, we ensure the examples among the training, validation and test sets have different source and translation sentences.

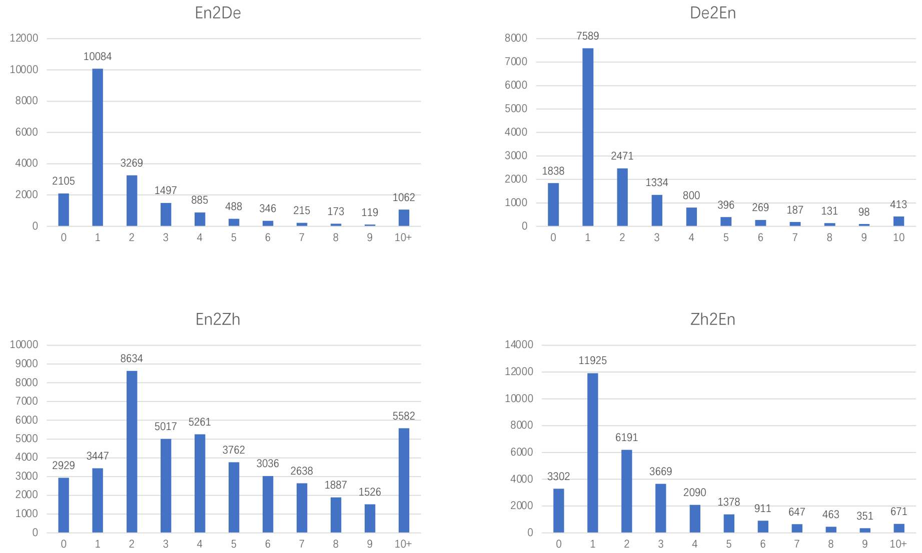


Figure 4: The length of the suggestion.

number of the incorrect spans with length 0 ranks top-2 among all the length buckets. This shows that under-translation is still a frequent error of the existing NMT models.

The length of the suggestions Figure 4 shows the length distribution of the suggestions. We can see that in English-to-German, German-to-English and Chinese-to-English, most of the suggestions contain only one word. For English-to-Chinese, most suggestions contain two Chinese characters. Additionally, we can also find that there are quite a few of suggestions with length zero in each translation direction. This shows that over-translation is a non-negligible problem for the existing NMT models.

4 Participants

Five participants submitted their systems to the subtask one of TS shared task. And two participants submitted their systems to the second sub-task. In sub-task one, 92 runs were submitted in total (each team is only allowed to submit less than 15 runs). Table 2 summarizes the participants and their affiliations.

4.1 Systems

Here we briefly describe each participant's systems as described by the authors and refer the reader to the participant's submission for further details. Since some participants did not submit their papers, we only describe the systems in the submitted papers.

Team	Institution
mind-ts	Soochow University and Alibaba
suda-hlt	Soochow University
Avocados	Beijing Jiaotong University
IOL Research	Transn IOL Technology CO., Ltd.
Slack	Zhejiang University

Table 2: The participating teams and their affiliations.

4.1.1 Baseline

We take the naive Transformer-base (Vaswani et al., 2017) as the baseline and directly apply the implementation of the open-source toolkit, fiarseq.[8] We construct the synthetic corpus based on the WMT parallel corpus, and we refer the readers for details about constructing the synthetic corpus in the paper (Yang et al., 2021). For training, we apply the two-state training pipeline, where we pre-train the model on the synthetic corpus in the first stage, and then fine-tune the model on the golden corpus in the second stage.

4.1.2 IOL Research

The team of IOL Research participates the two sub-tasks and focuses on the En-Zh and Zh-En translation directions. They use the ΔLM as their backbone model. ΔLM is a pre-trained multilingual encoder-decoder model, which outperforms various strong baselines on both natural language generation and translation tasks (Ma et al., 2021). Its encoder and decoder are initialized with the

[8] https://github.com/pytorch/fairseq

pre-trained multilingual encoder InfoXLM (Chi et al., 2020). Their model has 360M parameters, 12-6 encoder-decoder layers, 768 hidden size, 12 attention heads and 3072 FFN dimension. For the training data, they construct the synthetic data with two different methods according to its constructing complexity. During training, they use the two-stage fine-tuning, where they apply the synthetic data to fine-tune the original ΔLM in the first stage and then fine-tune the result of the first stage with the golden corpus. In their experiments, they find that the accuracy indicator of TS can be helpful for efficient PE in practice. Overall, they achieved the best scores on 3 tracks and comparable result on another track.

4.1.3 Avocados

The team of Avocados tries different model structures, such as Transformer-base (Vaswani et al., 2017), Transformer-big (Vaswani et al., 2017), SA-Transformer (Yang et al., 2021) and DynamicConv (Wu et al., 2019). They test different ensemble approaches for better performance. For more details, we refer the readers to their paper (Zhang et al., 2022). Their main efforts are paid on building the synthetic corpus. They apply three different ways to construct the synthetic corpus. Firstly, they randomly sample a sub-segment in each target sentence of the golden parallel data, mask the sampled sub-segment to simulate an incorrect span, and use the sub-segment as an alternative suggestion. Secondly, the same strategy as above is used for pseudo-parallel data with the target side substituted by machine translation results. Finally, they use a quality estimation model to estimate the translation quality of words in translation output sentence and select the span with low confidence for masking. Then, an alignment tool to find the sub-segment corresponding to the span in the reference sentence and use it as the alternative suggestion for the span. To bridge the domain difference between the large-scale synthetic data and human-annotated golden corpus, they apply the pre-trained BERT to filter data similar to the golden corpus as in-domain data, which are used as pre-training for the next phase after pre-training model with a large-scale synthetic corpus. Overall, they rank second and third on the English-German and English-Chinese bidirectional tasks respectively.

4.1.4 mind-ts

The team of mind-ts participate in the English-German and English-Chinese translation directions in the sub-task one, and their submissions are ranked first in three of four language directions. For English-German, they initialize the weights with NMT models released by teh winner of WMT19 (Ng et al., 2019). For English-Chinese, the one-to-many and many-to-one mBART50 models are used (Tang et al., 2020). Their main contribution is to construct the synthetic corpus with word alignment. They use the well-trained alignment models between source and target languages to filter out high-quality augment data. Specifically, they first use the Fast Align toolkit to extract the token alignments. Then, they remove tokens that appear in both MT and reference to get the trimmed result. They trim these common tokens because they want the model to focus more on the incorrect span and its alternative. Additionally, they use the dual conditional cross-entropy model to calculate the quality score of the pair between the source and masked translation sentences. If the cross-entropy quality score meets the threshold, they treat the masked translation and the alignment segments as the good examples for TS. Similarly, they also use the two-phase pre-training pipeline to get the final models.

4.2 Submission Summary

The submissions for this year's TS shared task cover different approaches from the pre-trained LMs and the encoder-decode NMT models. From the submissions, we find that the pre-trained models are very useful for the final performance. Additionally, almost all of the submissions have tried different approaches for constructing the synthetic corpus. As the amount of the golden corpus is limited, it is very important to find efficient ways to construct the synthetic corpus. The main problem for constructing synthetic corpus is how to make the synthetic corpus similar to the golden corpus in domain or other aspects. Finally, how to efficiently apply the synthetic corpus also needs much more efforts to investigate. All submissions adopt the two-stage training pipeline to train the models.

4.3 Evaluation Results

We report the BLEU scores of the submissions. The BLEU is calculated automatically with the sacrebleu toolkit. For each run, the participating team need to submit their top-1 suggestions for each sentence in the test set. Each participating team can submit at most 15 times for each track. We only report the best score for each team. Table 3 and 4 report the results on English-Chinese and English-German respectively in the sub-task one. Table 5 report the results on English-Chinese in the sub-task two.

Team	En-Zh	Zh-En
Baseline	31.02	25.84
mind-ts	33.92(2)	30.07 (1)
Avocados	33.33 (3)	28.56 (3)
IOL Research	39.71 (1)	28.42 (4)

Table 3: Evaluation results on the language pair for English-Chinese in the sub-task one. The number in bracket is the ranked position.

Team	En-De	De-En
Baseline	35.07	37.61
mind-ts	42.91(1)	47.04 (1)
Avocados	42.61 (2)	36.30 (2)

Table 4: Evaluation results on the language pair for English-German in the sub-task one. The number in bracket is the ranked position.

Team	En-Zh	Zh-En
Baseline	41.83	35.02
IOL Research	48.60 (1)	39.95 (1)

Table 5: Evaluation results on the language pair for English-Chinese in the sub-task two. The number in bracket is the ranked position.

5 Discussion and Analysis

Comparing the results of the BLEU scores of all submissions with our baseline systems, there is a significant gap between the submitted and baseline systems. This shows that there is a large space for us to try different techniques to improve the performance of TS. By comparing the results of different submitted systems, we find that different pre-training models have a large difference on the final performance. This is a similar trend with other NLP tasks. Therefore, we believe that this is an interesting and promising direction for us to pay much more efforts.

All submitted systems have investigated different approaches for constructing the synthetic corpus and almost all of them have achieved much improvements with the synthetic corpus. The noise in the synthetic corpus is a major problem which negatively affects the final performance. Therefore, how to filter or decrease the noise is an open question. The team of mind-ts applies the pre-trained LM to filter the synthetic corpus and obtain better performance on 3 out of 4 tracks based on the high-quality synthetic corpus. We can investigate more effective approaches to detect and filter the noise in the synthetic corpus.

However, another interesting direction which are not investigated by the submissions is modeling the interaction between the source and translation sentences efficiently. Compared to MT, the main difference for TS is that the input for TS is dual-source, namely the source and translation sentence. We believe that efficiently modeling the interaction between the source and translation sentences can improve the final performance.

6 Conclusion

We present the results of first edition of the Translation Suggestion shared task. For the goal of this task, we create and release the first golden benchmark dataset, called *WeTS*, which covers the language pairs for English-Chinese and English-German. We wish the released corpus can spur the researches in this area. This year we received 92 submissions from 5 participating teaming in the sub-task one and 6 submissions for the sub-task 2, most of them covering the two translation directions. Results of these submissions show that the pre-trained models and synthetic corpus are two important factors for the final performance.

Acknowledgements

We would like to thank Yaou Li and Ning Zhang for their helps on building the official website of the shared task. The authors would also like to thank the anonymous reviewers of this paper, and the anonymous reviewers of the previous version for their valuable comments and suggestions to improve our work.

References

Vicent Alabau, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Ulrich Germann, Jesús González-Rubio, Robin Hill, Philipp Koehn, Luis A Leiva, et al. 2014. Casmacat: A computer-assisted translation workbench. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 25–28.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, et al. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2020. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. *arXiv preprint arXiv:2007.07834*.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.

Spence Green, Jeffrey Heer, and Christopher D Manning. 2013. The efficacy of human post-editing for language translation. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 439–448.

Spence Green, Sida I Wang, Jason Chuang, Jeffrey Heer, Sebastian Schuster, and Christopher D Manning. 2014. Human effort and machine learnability in computer aided translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1225–1236.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Guoping Huang, Jiajun Zhang, Yu Zhou, and Chengqing Zong. 2015. A new input method for human translators: integrating machine translation effectively and imperceptibly. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Tomoyuki Kajiwara. 2019. Negative lexically constrained decoding for paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6047–6052.

Rebecca Knowles and Philipp Koehn. 2016. Neural interactive translation prediction. In *Proceedings of the Association for Machine Translation in the Americas*, pages 107–120.

Dongjun Lee, Junhyeong Ahn, Heesoo Park, and Jaemin Jo. 2021. IntelliCAT: Intelligent machine translation post-editing with quality estimation and translation suggestion. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 11–19, Online. Association for Computational Linguistics.

Huayang Li, Lemao Liu, Guoping Huang, and Shuming Shi. 2021. GWLAN: General word-level AutocompletioN for computer-aided translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4792–4802, Online. Association for Computational Linguistics.

Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, and Furu Wei. 2021. Deltalm: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders. *arXiv preprint arXiv:2106.13736*.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. *arXiv preprint arXiv:1907.06616*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Sebastin Santy, Sandipan Dandapat, Monojit Choudhury, and Kalika Bali. 2019. Inmt: Interactive neural machine translation prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 103–108.

Raymond Hendy Susanto, Shamil Chollampatt, and Liling Tan. 2020. Lexically constrained neural machine translation with Levenshtein transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3536–3543, Online. Association for Computational Linguistics.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Qian Wang, Jiajun Zhang, Lemao Liu, Guoping Huang, and Chengqing Zong. 2020. Touch editing: A flexible one-time interaction approach for translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 1–11.

Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. *arXiv preprint arXiv:1901.10430*.

Zhen Yang, Yingxue Zhang, Ernan Li, Fandong Meng, and Jie Zhou. 2021. Wets: A benchmark for translation suggestion. *arXiv preprint arXiv:2110.05151*.

Hongxiao Zhang, Siyu Lai, Songming Zhang, Hui Huang, Yufeng Chen, Jinan Xu, and Jian Liu. 2022. Improved data augmentation for translation suggestion. *arXiv preprint arXiv:2210.06138*.

Vilém Zouhar, Martin Popel, Ondřej Bojar, and Aleš Tamchyna. 2021. Neural machine translation quality and post-editing performance. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10204–10214, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Focused Concatenation for Context-Aware Neural Machine Translation

Lorenzo Lupo[1] **Marco Dinarelli**[1] **Laurent Besacier**[2]
[1]Université Grenoble Alpes, France
[2]Naver Labs Europe, France
lorenzo.lupo@univ-grenoble-alpes.fr
marco.dinarelli@univ-grenoble-alpes.fr
laurent.besacier@naverlabs.com

Abstract

A straightforward approach to context-aware neural machine translation consists in feeding the standard encoder-decoder architecture with a window of consecutive sentences, formed by the current sentence and a number of sentences from its context concatenated to it. In this work, we propose an improved concatenation approach that encourages the model to focus on the translation of the current sentence, discounting the loss generated by target context. We also propose an additional improvement that strengthen the notion of sentence boundaries and of relative sentence distance, facilitating model compliance to the context-discounted objective. We evaluate our approach with both average-translation quality metrics and contrastive test sets for the translation of inter-sentential discourse phenomena, proving its superiority to the vanilla concatenation approach and other sophisticated context-aware systems.

1 Introduction

While current neural machine translation (NMT) systems have reached close-to-human quality in the translation of decontextualized sentences (Wu et al., 2016), they still have a wide margin of improvement ahead when it comes to translating full documents (Läubli et al., 2018). Many works tried to reduce this margin, proposing various approaches to context-aware NMT (CANMT)[1]. A common taxonomy (Kim et al., 2019; Li et al., 2020) divides them in two broad categories: multi-encoding approaches and concatenation (single-encoding) approaches. Despite its simplicity, the concatenation approaches have been shown to achieve competitive or superior performance to more sophisticated, multi-encoding systems (Lopes et al., 2020; Ma et al., 2021). Nonetheless, it has been shown that Transformer-based NMT systems (Vaswani et al., 2017) struggle to learn locality properties (Hardmeier, 2012; Rizzi, 2013) of both the language itself and the source-target alignment when the input sequence grows in length, as in the case of concatenation (Bao et al., 2021). Unsurprisingly, the presence of context makes learning harder for concatenation models by distracting attention. Moreover, we know from recent literature that NMT systems require context for a sparse set of inter-sentential discourse phenomena only (Voita et al., 2019; Lupo et al., 2022). Therefore, it is desirable to make concatenation models more focused on local linguistic phenomena, belonging to the current sentence, while also processing its context for enabling inter-sentential contextualization whenever it is needed. We propose an improved concatenation approach to CANMT that is more focused on the translation of the current sentence by means of two simple, parameter-free solutions:

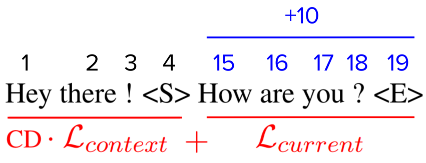

+10
1 2 3 4 15 16 17 18 19
Hey there ! <S> How are you ? <E>
CD · $\mathcal{L}_{context}$ + $\mathcal{L}_{current}$

Figure 1: Example of the proposed approach applied over a window of 2 sentences, with context discount CD and segment-shifted positions by a factor of 10.

- Context-discounting: a simple modification of the NMT loss that improves context-aware translation of a sentence by making the model less distracted by its concatenated context;
- Segment-shifted positions: a simple, parameter-free modification of position embeddings, that facilitates the achievement of the context-discounted objective by supporting the learning of locality properties in the document translation task.

We support our solutions with extensive experi-

[1] Unless otherwise specified, we refer to *context* as the sentences that precede or follow a *current* sentence to be translated, within the same document.

ments, analysis and benchmarking.

2 Background

2.1 Multi-encoding approaches

Multi-encoding models couple a self-standing sentence-level NMT system, with parameters θ_S, with additional parameters θ_C that encode and integrate the context of the current sentence, either on source side, target side, or both. The full context-aware architecture has parameters $\Theta = [\theta_S; \theta_C]$. Multi-encoding models differ from each other in the way they encode the context or integrate its representations with those of the current sentence. For instance, the representations coming from the context encoder can be integrated with the encoding of the current sentence outside the decoder (Maruf et al., 2018; Voita et al., 2018; Zhang et al., 2018; Miculicich et al., 2018; Maruf et al., 2019; Zheng et al., 2020) or inside the decoder (Tu et al., 2018; Kuang et al., 2018; Bawden et al., 2018; Voita et al., 2019; Tan et al., 2019), by making it attending to the context representations directly, using its internal representation of the decoded history as query.

2.2 Single-encoder approaches

The concatenation approaches are the simplest in terms of architecture, as they mainly consist in concatenating each (current) source sentence with its context before feeding it to the standard encoder-decoder architecture (Tiedemann and Scherrer, 2017; Junczys-Dowmunt, 2019; Agrawal et al., 2018; Ma et al., 2020), without the addition of extra learnable parameters. The decoding can then be limited to the current sentence, although decoding the full target concatenation is more effective thanks to the availability of target context. A typical strategy to train a concatenation approach and generate translations is by sliding windows (Tiedemann and Scherrer, 2017). An sKtoK model decodes the translation $\boldsymbol{y}_K^j$ of a source window $\boldsymbol{x}_K^j$, formed by K consecutive sentences belonging to the same document: the current (jth) sentence and $K-1$ sentences concatenated as source-side context. Besides the end-of-sequence token <E>, another special token <S> is introduced to mark sentence boundaries in the concatenation:

$$\boldsymbol{x}_K^j = \boldsymbol{x}^{j-K+1}{}_{\text{<S>}}\boldsymbol{x}^{j-K+2}{}_{\text{<S>}\ldots\text{<S>}}\boldsymbol{x}^{j-1}{}_{\text{<S>}}\boldsymbol{x}^{j}{}_{\text{<E>}}$$
$$\boldsymbol{y}_K^j = \boldsymbol{y}^{j-K+1}{}_{\text{<S>}}\boldsymbol{y}^{j-K+2}{}_{\text{<S>}\ldots\text{<S>}}\boldsymbol{y}^{j-1}{}_{\text{<S>}}\boldsymbol{y}^{j}{}_{\text{<E>}}$$

Both past and future contexts can be concatenated to the current pair $\boldsymbol{x}^j, \boldsymbol{y}^j$, although in this work we consider only the past context, for simplicity. At training time, the loss is calculated over the whole output $\boldsymbol{y}_K^j$, but only the translation $\boldsymbol{y}^j$ of the current sentence is kept at inference time, while the translation of the context is discarded. Then, the window is slid by one position forward to repeat the process for the $(j+1)$th sentence and its context. Concatenation approaches are trained by optimizing the same objective function as standard NMT over a window of sentences:

$$\mathcal{L}(\boldsymbol{x}_K^j, \boldsymbol{y}_K^j) = \sum_{t=1}^{|\boldsymbol{y}_K^j|} \log P(y_{K,t}^j | \boldsymbol{y}_{K,<t}^j, \boldsymbol{x}_K^j), \quad (1)$$

so that the likelihood of the current target sentence is conditioned on source and target context.

2.3 Closing the gap

Concatenation approaches have the advantage of treating the task of CANMT in the same way as context-agnostic NMT, which eases learning because the learnable parameters responsible for inter-sentential contextualization are the same that undertake intra-sentential contextualization. Indeed, learning the parameters responsible for inter-sentential contextualization in multi-encoding approaches (θ_C) has been shown to be challenging because the training signal is sparse and the task of retrieving useful context elements difficult (Lupo et al., 2022). Nonetheless, encoding current and context sentences together comes at a cost. In fact, when sequences are long the risk of paying attention to irrelevant elements increases. Paying attention to the "wrong tokens" can harm their intra and inter-sentential contextualization, associating them to the wrong latent features. Indeed, Liu et al. (2020) and Sun et al. (2022) showed that learning to translate long sequences, comprised of many sentences, fails without the use of large-scale pre-training or data-augmentation (e.g., like Junczys-Dowmunt (2019) and Ma et al. (2021) did). Bao et al. (2021) provided some evidence about this leaning difficulty, showing that failed models, i.e., models stuck in local minima with a high validation loss, present a distribution of attention weights that is flatter (with higher entropy), both in the encoder and the decoder, than the distribution occurring in models that converge to lower validation loss. In other words, attention struggles to learn the locality properties of both the language itself and the source-target alignment (Hardmeier, 2012; Rizzi, 2013). As a solution, Zhang et al. (2020) and Bao et al. (2021) propose two slightly different masking methods that allow both the encoding of the current sentence concatenated with context, and the separate encoding of each sentence in window. The representations generated by the two encoding schemes are then integrated together, at the cost of adding extra learnable parameters to the standard Transformer architecture.

3 Proposed approach

3.1 Context discounting

Evidently, Equation 1 defines an objective function that does not factor in the fact that we only care about the translation of the current sentence $\boldsymbol{x}^j$, because the context translation will be discarded during inference. Moreover, as discussed above, we need attention to stay focused locally, relying on context only for the disambiguation of relatively sparse inter-sentential discourse phenomena that are ambiguous at sentence level. Hence, we propose to encourage the model to focus on the translation of the current sentence $\boldsymbol{x}^j$ by applying a discount $0 \leq \text{CD} < 1$ to the loss generated by context tokens:

$$\begin{aligned}\mathcal{L}_{\text{CD}}(\boldsymbol{x}_K^j, \boldsymbol{y}_K^j) &= \text{CD}\cdot\mathcal{L}_{context} + \mathcal{L}_{current} \qquad (2)\\ &= \text{CD}\cdot\mathcal{L}(\boldsymbol{x}_{K-1}^{j-1}, \boldsymbol{y}_{K-1}^{j-1}) + \mathcal{L}(\boldsymbol{x}^j, \boldsymbol{y}^j).\end{aligned}$$

This is equivalent to consider an sKtoK concatenation approach as the result of a multi-task sequence-to-sequence setting (Luong et al., 2016), where an sKto1 model performs the *reference task* of translating the current sentence given a concatenation of its source with K-1 context sentences, while the translation of the context sentences is added as a secondary, complementary task. The reference task is assigned a bigger weight than the secondary task in the multi-task composite loss. As we will see in Section 4.5, this simple modification of the loss allows the model to learn a self-attentive mechanism that is less distracted by noisy context information, thus achieving net improvements in the translation of inter-sentential discourse phenomena occurring in the current sentence (Section 4.3), and helping concatenation systems to generalize to wider context after training (Section 4.5.3).

3.2 Segment-shifted positions

Context discounting pushes the model to discriminate between the current sentence and the context. Such discrimination can be undertaken by cross-referencing the information provided by two elements: sentence separation tokens <S>, and sinusoidal position encodings, as defined in (Vaswani et al., 2017). In order to facilitate this task, we propose to provide the model with extra information about sentence boundaries and their relative distance. (Devlin et al., 2019) achieve this goal by adding segment embeddings to every token representation in input to the model, on top of token and position embeddings, such that every segment embedding represents the sentence position in the window of sentences. However, we propose an alternative solution that does not require any extra learnable parameter nor memory allocation: segment-shifted positions. As shown in Figure 1, we apply a constant shift after every separation token <S>, so that the resulting token position is equal to its original position plus a total shift depending on the chosen constant *shift* and the index $k = 1, 2, ..., K$ of the sentence the token belongs to: $t' = t + k * shift$. As a result, the position distance between tokens belonging to different sentences is increased. For example, the distance between the first token of the current sentence and the last token of the preceding context sentence increases from 1 to 1 + *shift*. By increasing the distance between sinusoidal position embeddings[2] of tokens belonging to different sentences, their dot product, which is at the core of the attention mechanism, becomes smaller, possibly resulting in smaller attention weights. In other words, the resulting attention becomes more localized, as confirmed by the empirical analysis reported in Section 4.6.1. In Section 4.3, we present results of segment-shifted positions, and then compare them with both sinusoidal segment embeddings and learned segment embeddings in Section 4.6.2.

4 Experiments

4.1 Setup[3]

We conduct experiments with two language pairs and domains. For En→Ru, we adopt a document-level corpus released by Voita et al. (2019), based on OpenSubtitles2018 (with dev and test sets), comprised of 1.5M parallel sentences. For En→De, we train models on TED talks subtitles released by IWSLT17 (Cettolo et al., 2012). Models are tested

[2] Positions can be shifted by segment also in the case of learned position embeddings, both absolute and relative. We leave such experiments for future works.

[3] See Appendix A for more details.

on IWSLT17's test set 2015, while test-sets 2011-2014 are used for development, following related works in the literature.

Besides evaluating average translation quality with BLEU[4] (Papineni et al., 2002) and COMET[5] (Rei et al., 2020), we employ two contrastive test suites for the evaluation of the translation of inter-sentential discourse phenomena. For En→Ru, we adopt Voita et al. (2019)'s test suite for evaluation on deixis, lexical cohesion, verb-phrase ellipsis and inflection ellipsis. This test suite is comprised of a development set with examples of deixis and lexical cohesion, that we adopted for a preliminary analysis of context discounting. For En→De, we evaluate models on ambiguous pronoun translation with ContraPro (Müller et al., 2018), a large contrastive set of ambiguous pronouns whose antecedents belong to context. In order to validate the improvements achieved by our approaches on the test sets, we perform statistical significance tests, detailed in Annex A.1.

We experiment with two models: 1) **base**: a context-agnostic baseline following *Transformer-base* (Vaswani et al., 2017); 2) **s4to4**: a context-aware concatenation approach with the exact same architecture as *base*, but that adopts sliding windows of 4 concatenated sentences as source and target. An implementation of these models and the proposed approach can be found on github.[6]

4.2 Preliminary analysis

As a preliminary analysis, we evaluate the impact of various values of context discounting on the performance of concatenation approaches with sliding windows, in order to choose one value for all the subsequent experiments. We train En→Ru s4to4 models with context discounts ranging from 1 (no context discounting) to 0 (context loss is completely ignored): $\texttt{CD} = 1.0, 0.9, 0.7, 0.5, 0.3, 0.1, 0.01, 0$. We evaluate these models on the development sets by means of their average loss calculated over the current target sentence (*current loss*) and the average accuracy on the disambiguation of discourse phenomena. The results are plotted on Figure 2. We find out that the stronger the context discounting, the better the performance, with an improving trend from $\texttt{CD} = 1$ to $\texttt{CD} = 0.01$. Performance drops on the extreme case of $\texttt{CD} = 0$, likely because too much training signal is lost in this situation (all the training signal coming from the context is completely ignored). As such, we set $\texttt{CD} = 0.01$ for all of our following experiments.

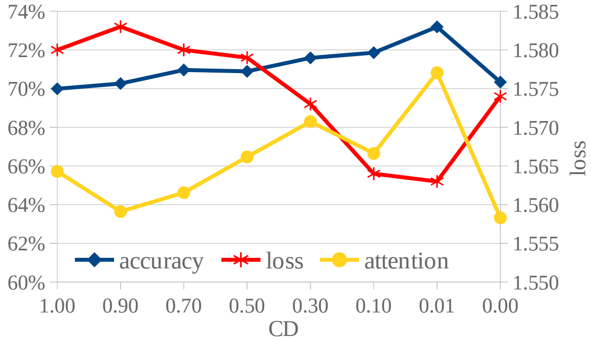


Figure 2: Evaluation of En→Ru s4to4 trained with various levels of context discounting, ranging from 1 to 0. We plot the best *current loss* obtained by each model on the development set (red), and its average accuracy on the development portion of the contrastive set on discourse phenomena (blue). In yellow, the average portion of attention that is focused on the current sentence (see Section 4.5.2).

4.3 Main results

Tables 1 and 2 display the main evaluation results measured in terms of accuracy on contrastive test sets (Disc.) and BLEU, for the En→Ru and En→De language pairs, respectively. We first observe that s4to4 is a strong context-aware baseline as it improves accuracy on contrastive sets by a large margin compared to the context-agnostic *base*, as already reported by previous works (Voita et al., 2019; Zhang et al., 2020; Lopes et al., 2020).

Average translation quality as measured by BLEU is virtually the same for all models. Indeed, our main focus is on contrastive evaluation of discourse translation, since average translation quality metrics like BLEU have been repeatedly shown to be ill-equipped to detect improvements in CANMT (Hardmeier, 2012). Learned average translation quality metrics like COMET might be more sensitive to inter-sentential discourse phenomena when applied at document-level, as we do. However, COMET differences are also negligible: all models perform on par according to statistical significance tests, except for the En→Ru model with context discount and segment shifting, that outperforms all the others with statistical significance.

When evaluating the accuracy on inter-sentential discourse phenomena, instead, we remark relevant

[4] Moses' *multi-bleu-detok* (Koehn et al., 2007) for De, *multi-bleu* for lowercased Ru as Voita et al. (2019).

[5] Default model: wmt20-comet-da.

[6] https://github.com/lorelupo/focused-concat

En→Ru system	Deixis	Lex co.	Ell. inf	Ell. vp	Disc.	BLEU	COMET
base	50.00	45.87	51.80	27.00	46.64	31.98	0.321
s4to4	85.80	46.13	79.60	73.20	72.02	32.45	0.329
s4to4 + CD	**87.16***	46.40	81.00	78.20*	73.42*	32.37	0.328
s4to4 + shift + CD	85.76	**48.33***	**81.40**	**80.40***	**73.55***	32.37	0.334*

Table 1: Accuracy on the En→Ru contrastive set for the evaluation of discourse phenomena (Disc., %), and BLEU score on the corresponding test set. The accuracy on Disc. is detailed on its left with the accuracy on each of the 4 discourse phenomena evaluated in the contrastive set. The symbol * denotes statistically significant ($p < 0.05$) improvements w.r.t. base and s4to4.

En→De system	$d = 1$	$d = 2$	$d = 3$	$d > 3$	Disc.	BLEU	COMET
base	32.89	43.97	47.99	70.58	37.27	29.63	0.546
s4to4	68.89	74.96	79.58	**87.78**	71.35	29.48	0.536
s4to4 + CD	**72.86***	75.96	80.10	84.38	74.31*	29.32	0.522
s4to4 + shift + CD	72.56*	**77.15***	**80.27**	86.65	**74.39***	29.20	0.528

Table 2: Accuracy on the En→De contrastive sets for the evaluation of discourse phenomena (Disc., %), and BLEU score on the corresponding test sets. The accuracy on Disc. is detailed on its left with the accuracy on anaphoric pronouns with antecedents at different distances $d = 1, 2, ...$ (in number of sentences). The symbol * denotes statistically significant ($p < 0.05$) improvements w.r.t. base and s4to4.

performance improvements. In fact, adding a 0.01 context discounting (+ CD) improves the accuracy on all of the 4 discourse phenomena under evaluation in En→Ru, and for all distances of pronoun's antecedents in En→De, with the sole exception of $d > 3$, proving to be an effective solution. Adding segment-shifted positions further improves performance for 3 discourse phenomena out of 4, and for pronouns with antecedents at distances $d = 1, 2$, showing that sliding windows systems often benefit from enhanced sentence position information in order to achieve the discounted CANMT objective. For both language pairs, we adopt a segment-shifting equal to the average sentence length, calculated over the entire training corpus, i.e., +8 positions for En→Ru and +21 positions for En→De. Experiments with other shifting values are reported in Section 4.6.3.

As a further experiment, we apply our solutions to concatenation models with concatenated windows shorter than 4 sentences,[7] and evaluate them in the En→Ru setting. The results presented in Table 3 show that context discounting is effective for s2to2 and s3to3 too, while adding segment-shifted positions only helps s2to2 + CD. As in the case of s4to4, BLEU only displays negligible fluctuations.

System	Disc.	BLEU
s2to2	59.10	32.73
s2to2 + CD	60.28*	32.69
s2to2 + shift + CD	**60.54***	32.41
s3to3	65.58	32.34
s3to3 + CD	**67.02***	32.42
s3to3 + shift + CD	66.98*	32.45

Table 3: Accuracy on the En→Ru contrastive set for the evaluation of discourse phenomena (Disc., %), and BLEU score on the test set. The symbol * denotes statistically significant ($p < 0.05$) improvements w.r.t. s2to2/s3to3. Our approach is effective for different concatenation windows.

4.4 Benchmarking

For a wider contextualization of our results, we compare in Table 4 our best system with other CANMT systems from the literature. For the En→Ru language pair, we compare with all the systems from the literature that were trained and evaluated under the same experimental conditions as ours, to the best of our knowledge. In particular, we report the results by Chen et al. (2021), Sun et al. (2022)' *MR Doc2Doc*, Zheng et al. (2020), Kang et al. (2020)'s *CADec + DCS-pf* and Zhang et al. (2020). All of them are sophisticated CANMT systems that add extra trainable parameters to the

[7] We cannot evaluate with more sentences because 4 is the maximum size of documents in the test sets specialized on discourse phenomena.

System	En→Ru					En→De				
	Deixis	Lex co.	Ell. inf	Ell. vp	Disc.	d=1	d=2	d=3	d>3	Disc.
Chen et al. (2021)	62.30	47.90	64.90	36.00	55.61	n.a.	n.a.	n.a.	n.a.	n.a.
Sun et al. (2022)	64.70	46.30	65.90	53.00	58.13	n.a.	n.a.	n.a.	n.a.	n.a.
Zheng et al. (2020)	61.30	58.10	72.20	80.00	63.30	n.a.	n.a.	n.a.	n.a.	n.a.
Kang et al. (2020)	79.20	62.00	71.80	80.80	73.46	n.a.	n.a.	n.a.	n.a.	n.a.
Zhang et al. (2020)	**91.00**	46.90	78.20	**82.20**	**75.61**	n.a.	n.a.	n.a.	n.a.	n.a.
Maruf et al. (2019)	n.a.	n.a.	n.a.	n.a.	n.a.	34.70	46.40	51.10	70.10	39.15
Voita et al. (2018)	n.a.	n.a.	n.a.	n.a.	n.a.	39.00	48.00	54.00	66.00	42.55
Stojanovski and Fraser (2019)	n.a.	n.a.	n.a.	n.a.	n.a.	53.00	46.00	50.00	71.00	52.55
Lupo et al. (2022)	n.a.	n.a.	n.a.	n.a.	n.a.	56.50	44.90	48.70	73.30	54.98
Müller et al. (2018)	n.a.	n.a.	n.a.	n.a.	n.a.	58.00	55.00	55.00	75.00	58.13
s4to4 + shift + CD (ours)	85.76	**48.33**	**81.40**	80.40	73.56	**72.56**	**77.15**	**80.27**	**86.65**	**74.39**

Table 4: Benchmarking: accuracy (%) on the contrastive sets for the evaluation of discourse phenomena (Disc., %).

Transformer architecture. Despite being the simplest and the only parameter free approach, our method outperforms all the others on lexical cohesion and noun phrase inflection based on elided context, while it is only second to Zhang et al. (2020) on deixis and verb-phrase ellipsis. BLEU scores were not available for comparison on the same test set, except for Zhang et al. (2020), which scored 31.84 BLEU points against the 32.45 BLEU points of our method.

For the En→De language pair, we compare to the literature performing evaluation on Müller et al. (2018)'s test set and providing details about their accuracy on pronouns with antecedents at $d > 1$. In particular: Maruf et al. (2019)'s best offline system, Stojanovski and Fraser (2019)'s *pron-25→pron-0**, Lupo et al. (2022)'s *K1-d&r*, Müller et al. (2018)'s *s-hier-to-2.tied* and their evaluation of Voita et al. (2018)'s architecture.[8] All of these works but Maruf et al. (2019) adopt the much larger WMT17[9] dataset for training. Despite this advantage, our system outperforms each of them on all the discourse phenomena under evaluation, by a large margin.

Notably, from this comparison it might seem that our approach is proposed in opposition to the others reported in Table 4, but it can actually be complimentary to many of them, such as (Zhang et al., 2020)'s, hopefully in a synergistic way. We encourage future research to investigate this possibility.

4.5 Analysis of context-discounting

4.5.1 Loss distribution

In this section, we analyze the impact of context discounting on the ability of the model to predict the translation of the current sentence. On the left side of Figure 3 we plotted the evolution along training epochs of the loss calculated on the current target sentence (*current loss*), for the En→Ru language pair. The right side, instead, represents the ratio between the *current loss* and the average loss-per-sentence calculated on the context sentences belonging to the same sliding window. These results support empirically our idea of context discounting as a solution to improve model performance on the current sentence. They also confirm that a strong discounting works best. Interestingly, predictions are improved on the current sentence (left) partially as a result of a trade-off with context quality (right). In fact, the current/context loss ratio of context-discounted models increases along training even when the *current loss* is decreasing, indicating that, at the beginning of training, context discounting pushes the model to only care about current predictions, but later it allows for good predictions of the context too. Such behavior is in line with the intuition that a good translation of the current sentence, even if strongly prioritized, also requires a good translation of the context. Otherwise, it is not possible to systematically solve the translation ambiguities referring to context.

4.5.2 Attention distribution

In this section we show some empirical evidence in favor of our intuition that context-discounting improves performance by helping the self-attentive mechanism to be more focused on the current sentence (less distracted by context). We analyzed

[8] Whenever the cited works present and evaluate multiple systems, we compare to the best performing one. To the best of our knowledge, we are including all the relevant works available in the literature. BLEU scores are not compared because, besides using different training data, the cited works don't adopt the same test set neither, with the sole exception of (Lupo et al., 2022).

[9] http://www.statmt.org/wmt17/translation-task.html

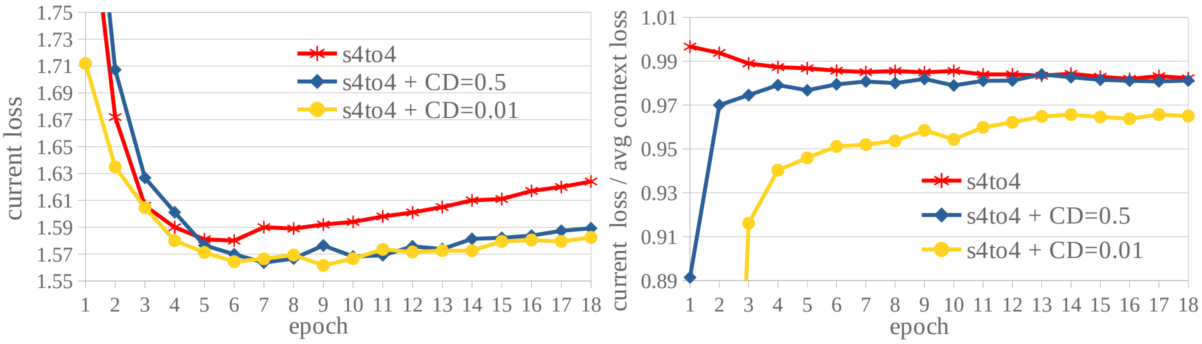


Figure 3: Context discounting enables better predictions of the current sentence (lower validation loss, on the left) at the expense of context sentences (lower current/context validation loss ratio, on the right). Language pair: En→Ru.

the distribution of the self-attention weights generated by the queries belonging to the current sentence (*current queries*), and how it is impacted by context discounting. Figure 2 clearly shows that context-discounting impacts the distribution of attention weights by skewing it towards the current sentence: a higher percentage of the total attention from *current queries* is directed towards tokens belonging to the (same) current sentence. As expected, the higher the context-discounting, the higher the portion of attention that is not dispersed towards context. The limit case of $CD = 0$ is not aligned with this trend, however. We suspect that the attention distribution is more flat in this case because the model encounters learning difficulties due to the training signal from the context being completely ignored (c.f. Bao et al. (2021) on non-fully-converged models having a flatter attention distribution).

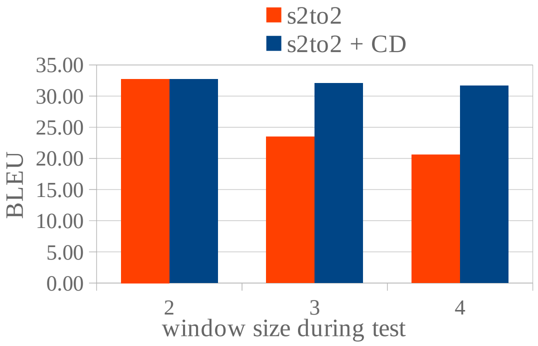


Figure 4: Our approach improves robustness of En→Ru s2to2 to window sizes unseen during training.

4.5.3 Robustness

Figure 4 shows that the s2to2 model is not robust to the translation of concatenation windows longer than those seen during training, i.e. longer than 2 sentences. Indeed, s2to2 loses 9.23 BLEU points when translating the same test set with windows of 3 sentences, and 12.14 BLEU points when translating with windows of 4. Instead, the context discounted model (blue bars) is very robust to unseen context lengths, being capable of translating them with minor degradation in average translation quality (-0.68 and -1.06 BLEU points for windows of 3 and 4, respectively). We observe a similar trend for s3to3, that loses 1.74 BLEU points when tested with windows of size 4, but recovers completely when equipped with context-discounting. The increased robustness of the concatenation models w.r.t. context size suggests once again that context discounting helps the models focusing on the current sentence.

4.6 Analysis of segment-shifted positions

4.6.1 Attention distribution

As a complementary evaluation, we tested if segment-shifted positions work as intended, i.e., by helping context-discounted models to learn the locality properties of both the language itself and the source-target alignment (Hardmeier, 2012; Rizzi, 2013). In other words, we expect segment-shifted positions to result in a more localized attention-distribution, in each of the sentences belonging to the concatenated sequence. To this aim, we computed the average entropy of the distribution of attention weights generated by all queries (both from current and context sentences), in both self and cross-attention. Results are shown in Table 5: context-discounting slightly reduces the average entropy, and this effect is amplified with the adoption of segment-shifted positions. Segment-shifted positions make attention more focused locally, as intended, which explains why the job of context

System	Attn entropy
s4to4	2.293
s4to4 + CD	2.276
s4to4 + shift + CD	**2.251**

Table 5: Average entropy of self and cross-attention weights decreases with the help of context-discounting and segment-shifted positions. All of the three values are different from one another with statistical significance (p<0.01).

	En→Ru		En→De	
System	Disc.	BLEU	Disc.	BLEU
s4to4 + shift + CD	73.56	32.45	**74.39**	29.20
s4to4 + lrn + CD	**73.68**	32.45	72.14	28.35
s4to4 + sin + CD	73.48	32.53	73.88	29.23

Table 6: Comparison between segment-shifted positions, learned segment embeddings and sinusoidal segment embeddings. Approaches are evaluated with accuracy on contrastive sets for the evaluation of discourse phenomena (Disc., %), and BLEU score on test sets. Differences across models are not statistically significant (p>0.05), except for s4to4+lrn+CD on En→De.

discounting is eased by this solution.

4.6.2 Comparison with segment-embeddings

In this section we compare our parameter-free approach to include explicit information on segment position (segment-shifted positions), with learned segment embeddings (Devlin et al., 2019), and sinusoidal segment embeddings. The latter are added to token and position embeddings at input, in the very same way as learned segment embeddings, with the only difference that their parameters are not learned but defined in the same way as sinusoidal position embeddings (Vaswani et al., 2017). In order to evaluate which approach helps best with context-discounting, we trained a context-discounted concatenation model with learned segment embeddings (s4to4+lrn+CD), and one with sinusoidal segment embeddings (s4to4+sin+CD), and compared them with s4to4+shift+CD. The results reported in Table 6 do not display any statistically significant differences across the three alternatives (p>0.05), except for learned embeddings, that underperform with statistical significance the other two variants on En→De. Instead, sinusoidal segment embeddings are competitive with segment-shifted positions on both language pairs. We leave a more in-depth analysis of segment-embeddings for concatenation approaches to future works.

System	Shift	Disc.	BLEU
s4to4 + shift + CD	100.00	73.46	32.41
s4to4 + shift + CD	avg-sequence	**73.86**	32.37
s4to4 + shift + CD	avg-corpus	73.56	32.45

Table 7: Accuracy on the En→Ru contrastive set for the evaluation of discourse phenomena (Disc., %), and BLEU score on the test set. Differences across models are not statistically significant (p>0.05).

4.6.3 Segment-shifting variants

In the experiments reported above, we always adopt a shifting value equal to the average sentence length calculated over the entire training corpus (avg-corpus), i.e., +8 positions for En→Ru, +21 positions for En→De. In this section we evaluate two alternative strategies for the selection of the shifting value: 1) applying a big shift of 100 units, one order of magnitude bigger than the average sentence length in the corpus (100); 2) applying a shifting value equal to the average sentence length of each window, calculated dynamically for each window of 4 concatenated sentences (avg-sequence). The results of this study are reported in Table 7. We do not observe relevant differences in average translation quality (BLEU) nor accuracy on the translation of discourse phenomena, and therefore confirm that the avg-corpus approach is a good alternative.

5 Conclusions

We presented a simple, parameter-free modification of the NMT objective for context-aware translation with sliding windows of concatenated sentences: context discounting. We analyzed the impact of our approach in the trade-off between current sentence predictions and context sentence predictions, showing that context discounting helps the model to focus on the current sentence, as intended. As a result, the concatenation model significantly improves its ability to disambiguate inter-sentential discourse phenomena, and becomes more robust to different context sizes. As an additional inductive bias towards locality, we equipped our model with segment-shifted positions, marking more explicitly the boundaries between sentences. This solution brings further improvements on targeted evaluation metrics. In the attempt of explaining the empirical functioning of the proposed solutions, we analysed their impact on the distribution of the attention weights, showing that they make it more focused and skewed towards the current sentence, as intended.

Limitations and future works

Our experiments are limited to the use case of short concatenated windows (up to 4 sentences). This is enough for capturing most of the ambiguous inter-sentential discourse phenomena, that usually span across a few sentences only (Müller et al., 2018; Voita et al., 2019; Lupo et al., 2022). However, recent works suggest that longer context windows might be helpful to increase average translation quality (BLEU) of concatenation approaches (Junczys-Dowmunt, 2019; Bao et al., 2021; Sun et al., 2022), and long-range discourse phenomena could be handled. We hope to investigate the impact of context discounting on longer sequences in future works. We also encourage to test the effectiveness of our approach on a wider range of data scenarios: from very limited document-level data to very abundant, including back translation (Ma et al., 2021) and monolingual pre-training techniques (Junczys-Dowmunt, 2019; Sun et al., 2022), to understand whether these methods are only alternative to context discounting or there exist synergies. Furthermore, experimenting with future context is also needed (c.f. Wong et al. (2020)).

Acknowledgements

We thank the anonymous reviewers for their insightful comments. This work has been partially supported by the Multidisciplinary Institute in Artificial Intelligence MIAI@Grenoble Alpes (ANR-19-P3IA-0003).

References

Ruchit Rajeshkumar Agrawal, Marco Turchi, and Matteo Negri. 2018. Contextual Handling in Neural Machine Translation: Look Behind, Ahead and on Both Sides. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 11–20, Alacant, Spain.

Guangsheng Bao, Yue Zhang, Zhiyang Teng, Boxing Chen, and Weihua Luo. 2021. G-transformer for document-level machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3442–3455, Online. Association for Computational Linguistics.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1304–1313, New Orleans, Louisiana. Association for Computational Linguistics.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy. European Association for Machine Translation.

Linqing Chen, Junhui Li, Zhengxian Gong, Boxing Chen, Weihua Luo, Min Zhang, and Guodong Zhou. 2021. Breaking the corpus bottleneck for context-aware neural machine translation with cross-task pre-training. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2851–2861, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Christian Hardmeier. 2012. Discourse in Statistical Machine Translation. A Survey and a Case Study. *Discours. Revue de linguistique, psycholinguistique et informatique. A journal of linguistics, psycholinguistics and computational linguistics*, (11). 00039 Number: 11 Publisher: Presses universitaires de Caen.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233, Florence, Italy. Association for Computational Linguistics.

Xiaomian Kang, Yang Zhao, Jiajun Zhang, and Chengqing Zong. 2020. Dynamic context selection for document-level neural machine translation via reinforcement learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2242–2254, Online. Association for Computational Linguistics.

Yunsu Kim, Duc Thanh Tran, and Hermann Ney. 2019. When and why is document-level context useful in neural machine translation? In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 24–34, Hong Kong, China. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the*

2004 Conference on Empirical Methods in Natural Language Processing, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Shaohui Kuang, Deyi Xiong, Weihua Luo, and Guodong Zhou. 2018. Modeling coherence for neural machine translation with dynamic and topic caches. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 596–606, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796, Brussels, Belgium. Association for Computational Linguistics.

Bei Li, Hui Liu, Ziyang Wang, Yufan Jiang, Tong Xiao, Jingbo Zhu, Tongran Liu, and Changliang Li. 2020. Does multi-encoder help? a case study on context-aware neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3512–3518, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

António Lopes, M. Amin Farajian, Rachel Bawden, Michael Zhang, and André F. T. Martins. 2020. Document-level neural MT: A systematic comparison. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 225–234, Lisboa, Portugal. European Association for Machine Translation.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Lorenzo Lupo, Marco Dinarelli, and Laurent Besacier. 2022. Divide and Rule: Effective Pre-Training for Context-Aware Multi-Encoder Translation Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4557–4572, Dublin, Ireland. Association for Computational Linguistics.

Shuming Ma, Dongdong Zhang, and Ming Zhou. 2020. A simple and effective unified encoder for document-level machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3505–3511, Online. Association for Computational Linguistics.

Zhiyi Ma, Sergey Edunov, and Michael Auli. 2021. A Comparison of Approaches to Document-level Machine Translation. *arXiv:2101.11040 [cs]*. ArXiv: 2101.11040.

Sameen Maruf, André F. T. Martins, and Gholamreza Haffari. 2018. Contextual neural model for translating bilingual multi-speaker conversations. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 101–112, Brussels, Belgium. Association for Computational Linguistics.

Sameen Maruf, André F. T. Martins, and Gholamreza Haffari. 2019. Selective attention for context-aware neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3092–3102, Minneapolis, Minnesota. Association for Computational Linguistics.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157. 03511.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954, Brussels, Belgium. Association for Computational Linguistics.

Mathias Müller, Annette Rios, Elena Voita, and Rico Sennrich. 2018. A large-scale test set for the evaluation of context-aware pronoun translation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 61–72, Brussels, Belgium. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the*

40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing Neural Networks by Penalizing Confident Output Distributions. *arXiv:1701.06548 [cs]*. 00464 arXiv: 1701.06548.

Martin Popel and Ondřej Bojar. 2018. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70. 00112 arXiv: 1804.00247.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Stefan Riezler and John T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, Ann Arbor, Michigan. Association for Computational Linguistics.

Luigi Rizzi. 2013. Locality. *Lingua*, 130:169–186.

Dario Stojanovski and Alexander Fraser. 2019. Improving anaphora resolution in neural machine translation using curriculum learning. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 140–150, Dublin, Ireland. European Association for Machine Translation.

Zewei Sun, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Shujian Huang, Jiajun Chen, and Lei Li. 2022. Rethinking Document-level Neural Machine Translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3537–3548, Dublin, Ireland. Association for Computational Linguistics.

Xin Tan, Longyin Zhang, Deyi Xiong, and Guodong Zhou. 2019. Hierarchical modeling of global context for document-level neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1576–1585, Hong Kong, China. Association for Computational Linguistics.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark. Association for Computational Linguistics.

Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. 2018. Learning to remember translation history with a continuous cache. *Transactions of the Association for Computational Linguistics*, 6:407–420.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1198–1212, Florence, Italy. Association for Computational Linguistics.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274, Melbourne, Australia. Association for Computational Linguistics.

KayYen Wong, Sameen Maruf, and Gholamreza Haffari. 2020. Contextual neural machine translation improves translation of cataphoric pronouns. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5971–5978, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv:1609.08144 [cs]*. 00000 arXiv: 1609.08144.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542, Brussels, Belgium. Association for Computational Linguistics.

Pei Zhang, Boxing Chen, Niyu Ge, and Kai Fan. 2020. Long-short term masking transformer: A simple but effective baseline for document-level neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1081–1087, Online. Association for Computational Linguistics.

Zaixiang Zheng, Xiang Yue, Shujian Huang, Jiajun Chen, and Alexandra Birch. 2020. Towards making the most of context in neural machine translation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3983–3989. ijcai.org.

A Details on experimental setup

All models are implemented in *fairseq* (Ott et al., 2019) and follow the *Transformer-base* architecture (Vaswani et al., 2017): hidden size of 512, feed forward size of 2048, 6 layers, 8 attention heads, total 60.7M parameters. They are trained on 4 Tesla V100, with a fixed batch size of approximately 32k tokens for En→Ru and 16k for En→De. As it has been shown that Transformers need a large batch size for achieving the best performance (Popel and Bojar, 2018). We stop training after 12 consecutive non-improving validation steps (in terms of loss on dev), and we average the weights of the 5 checkpoints that are closest to the best performing checkpoint, included. We train models with the optimizer configuration and learning rate (LR) schedule described in Vaswani et al. (2017). The maximum LR is optimized for each model over the search space $\{7e-4, 9e-4, 1e-3, 3e-3\}$. The LR achieving the best loss on the validation set after convergence was selected. We use label smoothing with an epsilon value of 0.1 (Pereyra et al., 2017) for all settings. We adopt strong model regularization (dropout=0.3) following Kim et al. (2019) and Ma et al. (2021). At inference time, we use beam search with a beam of 4 for all models. We adopt a length penalty 0.6 for all models. The other hyperparameters were set according to the relevant literature (Vaswani et al., 2017; Popel and Bojar, 2018; Voita et al., 2019; Ma et al., 2021; Lopes et al., 2020).

A.1 Statistical hypothesis tests

We perform statistical hypothesis testing with McNemar's test McNemar (1947) for comparing accuracy results on the contrastive test sets. For comparing BLEU performances and mean entropy (Table 5), we use approximate randomization (Riezler and Maxwell, 2005) with 10000 and 1000 permutations, respectively. For COMET, the official library[10] has a built in tool for the calculation of statistical significance with Paired T-Test and bootstrap resampling (Koehn, 2004).

B Details on experimental results

In this section, we report more details about the results presented in our Tables.

B.1 Evaluation of the translation of discourse phenomena

For each model that we evaluated by its accuracy on the contrastive sets for the evaluation of discourse phenomena (Disc., %), we include in Table 8 the accuracy achieved on the different subsets of the contrastive sets, as already done for Tables 1, 2 and 4. For the En→Ru set (Voita et al., 2019), we report the accuracy on each of the 4 discourse phenomena under evaluation; for the En→De test set (Müller et al., 2018), the accuracy on anaphoric pronouns with antecedents at different distances $d = 1, 2, ...$ (in number of sentences). As it can be noticed, our approach mostly outperform baselines and other variants on the majority of the evaluation subsets. We also include the column Disc$_{\text{avg}}$, which is calculated, for both language pairs, as the average of the 4 columns before the vertical dashed line.

$$Disc. = \frac{d1 * 7075 + d2 * 1510 + d3 * 573 + (d > 3) * 442}{9600},$$

$$Disc_{\text{avg}} = \frac{d1 + d2 + d3 + d > 3}{4}.$$

Disc$_{\text{avg}}$ represents the average accuracy on the disambiguation of the discourse phenomena present in the contrastive sets, as if they were all present with the same frequency. Instead, Disc. represents the overall accuracy on the contrastive set, which is equivalent to the average over the same 4 columns, but weighted by the sample size (last row) of each penomenon represented by the columns. While Disc. is a proxy of the ability to correctly translate a distribution of inter-sentential discourse phenomena as represented in the contrastive set, Disc$_{\text{avg}}$ is a proxy for the average ability to translate each of the inter-sentential phenomena under evaluation. Interestingly, Disc$_{\text{avg}}$ captures more evidently than Disc. the improvement achieved by adding segment-shifted positions to the context-discounted concatenation models. Finally, Disc$_{all-d}$ is calculated like Disc. but it also take into account pronouns whose antecedent belong to the same sentence ($d = 0$, i.e., they don't require context).

[10] https://github.com/Unbabel/COMET

	En→Ru						En→De							
System	Deixis	Lex co.	Ell. inf	Ell. vp	Disc.	Disc$_{\text{avg}}$	d=0	d=1	d=2	d=3	d>3	Disc.	Disc$_{\text{avg}}$	Disc$_{all-d}$
base	50.00	45.87	51.80	27.00	46.64	47.67	68.75	32.89	43.97	47.99	70.58	37.27	48.86	43.57
s4to4	85.80	46.13	79.60	73.20	72.02	71.18	75.20	68.89	74.96	79.58	**87.78**	71.35	77.80	72.12
s4to4 + CD	**87.16**	46.40	81.00	78.20	73.42	73.19	**76.66**	**72.86**	75.96	80.10	84.38	74.31	78.33	**74.78**
s4to4 + shift + CD	85.76	**48.33**	**81.40**	**80.40**	**73.56**	**73.97**	75.25	72.56	**77.15**	**80.27**	86.65	**74.39**	**79.16**	74.56
s4to4 + sin + CD	87.96	**46.80**	78.00	**76.60**	73.48	72.34	**76.75**	**71.83**	**76.82**	**80.97**	**87.55**	**73.88**	**79.29**	**74.46**
s4to4 + lrn + CD	**88.12**	46.47	**81.20**	75.60	**73.68**	**72.85**	73.91	70.21	75.29	77.66	85.06	72.14	77.06	72.49
s4to4 + 100 + CD	**85.60**	**48.73**	**80.80**	**79.60**	**73.46**	**73.68**	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
s4to4 + avg-seq + CD	84.84	46.20	77.60	73.00	71.34	70.41	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
s2to2	61.84	46.07	74.60	69.00	59.10	62.88	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
s2to2 + CD	**62.88**	46.27	78.00	**71.60**	60.28	64.69	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
s2to2 + shift + CD	62.60	**46.60**	**81.20**	71.40	**60.54**	**65.45**	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
s3to3	73.52	45.87	78.00	72.60	65.58	66.45	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
s3to3 + CD	73.88	**46.80**	**82.40**	**78.00**	**67.02**	**67.45**	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
s3to3 + shift + CD	**75.24**	46.07	79.40	76.00	66.98	68.45	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
Chen et al. (2021)	62.30	47.90	64.90	36.00	55.61	52.78	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
Sun et al. (2022)	64.70	46.30	65.90	53.00	58.13	57.48	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
Zheng et al. (2020)	61.30	58.10	72.20	80.00	63.30	67.90	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
Kang et al. (2020)	79.20	62.00	71.80	80.80	73.46	73.45	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
Zhang et al. (2020)	**91.00**	46.90	78.20	**82.20**	**75.61**	**74.58**	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
(Maruf et al., 2019)	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	68.60	34.70	46.40	51.10	70.10	39.15	50.58	45.04
(Müller et al., 2018)	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	75.00	39.00	48.00	54.00	66.00	42.55	51.75	49.04
(Stojanovski and Fraser, 2019)	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	74.00	53.00	46.00	50.00	71.00	52.55	55.00	56.84
(Lupo et al., 2022)	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	**81.10**	56.50	44.90	48.70	73.30	54.98	55.85	**60.21**
(Müller et al., 2018)	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	65.00	**58.00**	**55.00**	**55.00**	**75.00**	**58.13**	**60.75**	59.51
Sample size	2500	1500	500	500	5000	5000	2400	7075	1510	573	442	9600	9600	12000

Table 8: Accuracy on contrastive sets for the evaluation of discourse phenomena (Disc., %) and on their subsets: for En→Ru, the accuracy on each of the 4 discourse phenomena under evaluation; for En→De, the accuracy on anaphoric pronouns with antecedents at different distances $d = 1, 2, ...$ (in number of sentences). Disc$_{all-d}$, includes also $d = 0$. Disc$_{\text{avg}}$ denotes the average of the 4 accuracies before the dashed line.

Does Sentence Segmentation Matter for Machine Translation?

Rachel Wicks[1] and **Matt Post**[1,2]
[1]Center for Language and Speech Processing
[2]Human Language Technology Center of Excellence
Johns Hopkins University
{rewicks@,post@cs.}jhu.edu

Abstract

For the most part, NLP applications operate at the sentence level. Since sentences occur most naturally not on their own but embedded in documents, they must be extracted and segmented via the use of a segmenter, of which there are a handful of options. There has been some work evaluating the performance of segmenters on intrinsic metrics, that look at their ability to recover human-segmented sentence boundaries, but there has been no work looking at the effect of segmenters on downstream tasks. We ask the question, “does segmentation matter?" and attempt to answer it on the task of machine translation. We consider two settings: the inference scenario, where sentences are passed into a black-box system whose training segmentation is mostly unknown, and the training setting, where researchers have full control over the process. We find that the choice of segmenter largely does not matter, so long as its behavior is not one of extreme under- or over-segmentation. For such settings, we provide some qualitative analysis examining their harms, and point the way towards document-level processing.

1 Introduction

Contemporary machine translation assumes a sentence-level paradigm. However, data doesn’t exist naturally at the sentence level, requiring the use of automatic segmenters to split the data at both training and inference time. Training data is prepared with the use of sentence segmenters,[1] which are preprocessing steps that occur prior to alignment and bitext creation. At test time, deployed models also require the use of a segmenter. Many times, for downloaded models, especially, this inference-time application must be made without knowing what segmenter was used to train the model, introducing a potential misalignment or discrepancy and resulting performance degradation.

Sentence segmentation itself has received only a little attention in the research literature, although there has been a recent uptick (Moore, 2021; Wicks and Post, 2021). But to our knowledge, no work has been done investigating the effects of segmentation on machine translation. In fact, most research papers do not deal with the question at all, relying as they do on pre-segmented parallel data for both training and test time. This is a practical problem for deployment scenarios, where segmentation must be considered. It is also a deeper problem, since segmentation is ultimately a modeling decision that should be noted and made available with any published models, such as is done for other modeling decisions affecting input text, such as normalization, tokenization, and subword processing.

To understand whether and to what extent segmentation matters, we ask a series of questions: (1) What segmenter is best used at inference time? (2) When training a model, how important is the choice of segmenter? We break down this last question into two settings: (i) the standard training procedure in which sentences from parallel documents are aligned (Gale and Church, 1993), and (ii) more recent “mining” approaches, which use sentence representations to find sentence pairs without regard for document boundaries.

We find that

- for two black-box models trained with unknown segmentation, inference-time segmentation largely does not matter;
- when training new models, more aggressive segmentation generally produces better models, but these models are less robust to training-/inference-time segmentation mismatch;
- Global bitext mining approaches generally outperform document-based alignment tech-

[1] Sometimes called *sentence breakers*.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 843 - 854
December 7-8, 2022 ©2022 Association for Computational Linguistics

niques, but the latter is more robust to under-segmented data at inference.

2 Evaluation

Our research questions address two scenarios. In the first, a researcher has downloaded a shared model and wishes to use it to translate new data. In many instances—perhaps most—the providers of the model have neither shared nor reported what segmenter they used. Likely the model was trained on "standard" provided datasets such as those from WMT. We would like to have some understanding of the effect of different segmentations when we don't have control over the training segmentation.

Alternatively, we have a "glass box" model. In this setting, we are training the model, and have full flexibility over the choice of segmentation. By reconstructing the entire NMT pipeline with segmentation as the first step of dataset preprocessing, the researcher has complete control over the resulting model. This settings provides us with a more granular look at the effect of segmenter choice.

2.1 Metric Settings

In order to evaluate in either of these settings, we need to address a difficulty: automatic metrics for machine translation, whether source-based or reference-based, compare the machine translation output for sentences on a *pre-segmented*. For example, the WMT20 en-de test set (Barrault et al., 2020) has 1,418 pre-segmented sentences,[2]. In order to evaluate the effect of segmenters, we need to run three steps:

1. Remove the provided segmentation
2. Re-segment and translate
3. Align the translation outputs to the original references

This alignment step is necessary because metric scores cannot be compared across different reference segmentations. And it is complicated because we have no guarantee that the new segmentation will line up cleanly with the existing one.

We address this problem with three different alignment approaches.

Preserve keeps the provided segmentation, skipping step (1) above. Segmenters are applied to each sentence separately. It is easy to restore the original segmentation by simply keeping track of the number of sub-splits that were created with each line. On the downside, it does not allow the segmenter its full flexibility.

Document is possible when the sentences of a test set are grouped into documents. In this setting, step (1) above is done, but only at the document level. The segmenter is applied to the sentences in each document. Step (3) is undertaken by treating each document as a single line. Because of this, the number of references changes, and numbers computed from this approach cannot be compared to the other two.

Realign provides full flexibility to each segmenter. For step (3), we concatenate all outputs, and then align its words to the original reference segmentation using a search algorithm described in Section 2.2.

As many of the test sets originate from news articles and include header information (which typically includes a designated line break), we additionally insert sentence-final punctuation where it is not provided. This allows all segmenters to recover this segmentation.

2.2 Aligning outputs to references

Assume a source sequence (S) comprising tokens $(s_1, s_2, s_3, ...s_n)$, which aligns to reference r_i and a subsequent source sequence (T) comprised of $(t_1, t_2, t_3, ...t_m)$, which aligns to reference r_j. In the released test set, there exists an explicit segmentation between s_n and t_1. If we maintain this segmentation, the realignment of the translated tokens is obvious: any sub-sequence spawned from S aligns to r_i and we can re-concatenate the translations for scoring.

However, in production, there isn't an explicit segmentation between these tokens. Therefore, to give the segmenters the full degree-of-freedom that one would find in production, we must remove these segmentations. In this scenario, a segmenter may create the subsequences of (s_1, s_2, s_3), $(s_4, ...s_n, t_1, t_2)$, and $(t_3, t_4, ...t_m)$. Translation can also re-order tokens which makes the realignment non-obvious.

The realignment can be reduced to a search problem. To limit the search, we can impose hard constraints on the alignment based on subsequence matching. The field of Biomedical Engineering has a similar problem when trying to align two similar (but not identical) DNA sequences. We

[2] In en-de the "sentences" are typically several sentences to promote document translation

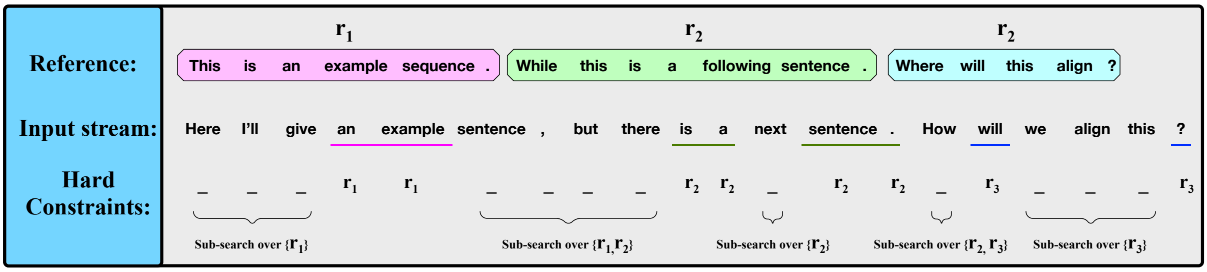

Reference:
Input stream:
Hard Constraints:
r1
r2
r2
This is an example sequence .
While this is a following sentence .
Where will this align ?
Here I'll give an example sentence , but there is a next sentence . How will we align this ?
_ _ _ r1 r1 _ _ _ _ r2 r2 _ r2 r2 _ r3 _ _ _ r3
Sub-search over {r1}
Sub-search over {r1,r2}
Sub-search over {r2}
Sub-search over {r2, r3}
Sub-search over {r3}

Figure 1: Example of the realignment method. Top row indicates the reference with grouped tokens each belonging to r_1, r_2, and r_3. The hard constraints are determined via longest subsequence matching (indicated with underlines). Note that not all matching surface forms may be determined as hard constraints based on token ordering. These hard constraints fix certain alignment points so the search algorithm (described in Section 2.2) has a limited reference set.

use an off-the-shelf capability[3] which maximizes subsequence matching length. This aligns some tokens to references so we search *between* the already aligned tokens. This is further illustrated in Figure 1.

Between a start and end token (t_i and t_j respectively) that are aligned to two references (r_x and r_y), we search for the best alignment of all intermediate tokens (t_k such that $i < k < j$) to a reference (r_z such that $x < z < y$). Plainly, this maintains a monotonicity: subsequent tokens can only be aligned to the same or a future reference.

We additionally require alignments to be consecutive sequences–no produced alignment to a reference can be a subsequence of an alignment to a different reference.

We optimize the alignment via the following costs:

- **Length-Ratio:** An optimal alignment should be the same length as the reference. This feature is the ratio of the shorter sequence to the longer sequence.
- **Final Punctuation:** A binary feature that determines if the aligned sentence and the original reference both end in punctuation.
- **N-gram Probability:** For unigrams and bigrams, the $p(t_k|r_z)$ or $p(t_{k-1}, t_k|r_z)$, respectively.
- **Start Word:** A binary feature that determines if the aligned sentence and the original reference both start with the same word.
- **End Word:** A binary feature that determines if the aligned sentence and the original reference both end with the same word.
- **Initial Capitalization:** A binary feature that determines if the aligned sentence and the original reference both start with a capitalized word.

We let the alignment cost be:

$$a_k = \sum_{i=0}^{k} a_i + \sum_{j} w_j * f_j(\text{alignment of } t_k)$$

where w_j is an associated weight on feature f_j. We perform a beam search with these features, expanding with each token t_k.

We use this methodology to re-align translations to references when the original segmentations are not maintained. We also note that this technique and toolkit can be used to reproduce alignments in other fields when the model's segmentation is not identical to that of the test sets as one might see in speech translation.

3 Experimental Setup

We focus on our investigation on English and German. We make this choice because this language pair has sufficient document-level information in datasets released by WMT. For many language pairs, datasets with true document pairs do not exist. A wider consideration of language pairs is not possible without further work to cultivate document-pair datasets.

Given document pairs in German and English, we extract sentences with a variety of segmenters and apply a typical document-based aligner to create bitext. Each segmenter creates a unique training set which we use to train a neural machine translation model.

Traditional alignment methodologies assume true document pairs. A search through both the source and target assumes alignments will be found

[3] https://biopython.org

in roughly sentence order. Vecalign (Thompson and Koehn, 2019) is an example of one of these document-based aligners. This method has the benefit of being capable of recovering erroneous segmentation because over-segmented sequences will still be consecutive during the search.

The growing field of bitext alignment has created new trends that search for sentence pairs outside the context of a document. One method of extracting sentences from all documents and searching globally for a sentence pair has created massive datasets such as CCMatrix (Schwenk et al., 2021b), WikiMatrix (Schwenk et al., 2021a), and CCAligned (El-Kishky et al., 2020). To compare the effects of segmentation in conjunction with both alignment techniques, we train models on data produced from all segmenters using both a document-based alignment method and a global, context-less based aligner.

3.1 Data

German–English has three datasets that preserve document-level boundaries in German-English—Europarl v10.[4] News Commentary v16.[5] and DGT[6] available through OPUS (Tiedemann, 2012). We find these datasets sufficient to train NMT models without other supplementary data.

Europarl comes from proceedings of the European Parliament. Aligned sentences are released as well as document IDs. The aligned sentences are roughly sentence-level. News Commentary is similarly produced from news articles.

DGT is a set of manually produced translations released by the European Commission's Directorate-General for Translation (DGT) from their translation memory. This dataset has substantial over-segmentation where one clause or phrase may be segmented onto its own line.

We use the Workshop on Machine Translation 2020 (WMT20) news task test sets and sacreBLEU[7] (Post, 2018) to score.

The sizes of the data before and after segmentation are available in Table 6 in the Appendix.

3.2 Segmentation models

We compare the following segmenters:

- **ORIGINAL:** The provided segmentations.

- **ALWAYS:** An over-segmentation approach that treats every piece of potentially sentence-ending punctuation as unambiguous.

- **ERSATZ:** A neural model that uses context windows to produce segmentations (Wicks and Post, 2021).

- **MOSES:** Always splits on punctuation, unless the previous token is in a pre-defined list of acronyms and abbreviations (Koehn et al., 2007).

- **PUNKT:** An unsupervised approach that uses thresholding to produce segmentations based on features such as casing, token length, and word frequency (Kiss and Strunk, 2006).

- **SPACY:** A "Rule-based" technique that varies on language.[8]

- **PAIRS:** The DGT dataset is oversegmented, and many lines contain less than one whole sentence. Lines must be merged in order to have complete sentences. In this setting, we merge the original bitext (instead of inserting segmentations). To implement, we simply combine every two lines together and treat as one "sentence." This merging also adds many-to-many sentence alignments for training in the Europarl and News Commentary datasets. *We only consider this "segmentation" method at training as the test data is sufficiently undersegmented.*

4 Segmentation at Inference with a Black Box System

In order to replicate a real-world use-case, we use an off-the-shelf pre-trained model. Datasets used to train these models are reported, but for the most-part, segmentation is unknown. We consider test sets in a variety of language pairs[9] for comprehensiveness. For model consistency, we chose a multilingual NMT model. We use the Prism (Thompson and Post, 2020) as a blackbox translation model,

After applying the segmentation methods described in Section 2, we translate with Prism.

[4] `https://www.statmt.org/europarl/v10/training/`

[5] `https://data.statmt.org/news-commentary/v16/training/`

[6] `https://opus.nlpl.eu/DGT.php`

[7] We considered COMET (Rei et al., 2020) as an alternative but did not find significant differences in trends

[8] v2.3.5, `https://spacy.io`

[9] `cs-en`, `de-en`, `en-cs`, `en-de`, `en-pl`, `en-ru`, `en-zh`, `pl-en`, `ru-en`, `zh-en`

	PRESERVE	REALIGN	DOCUMENT
	PRISM	PRISM	PRISM
ORIGINAL	27.1	27.5	28.9
ALWAYS	28.7	29.1	30.5
ERSATZ	29.0	29.4	30.8
MOSES	29.0	29.5	30.8
PUNKT	29.0	29.4	30.8
SPACY	28.7	29.3	30.6

Table 1: PRESERVE maintains original segmentations before applying the segmenter and aligns all produced sentences to the original corresponding reference. DOCUMENT removes segmentations from the original source before applying segmenter and aligns all translations to a single reference sentence (the entire document). REALIGN removes original segmentations and applies the alignment technique in Section 2.2 before scoring. Note that columns are not directly comparable. Differences between segmenters are not statistically significant.

We report results in Table 1 by averaging BLEU scores across the languages. As shown in the table, different segmenters are consistent within each alignment technique. The original test sets from some language pairs (namely `cs-en`, `en-cs`, `de-en`, and `en-de`) were undersegmented in the release to encourage document-level MT. For this reason, the average with the ORIGINAL segmentation is lower—the Prism model trained primarily on sentence-level data does not generalize as well to multiple sentence inputs. PRESERVE and REALIGN have the same number of references while DOCUMENT doesn't. PRESERVE and REALIGN are more directly comparable but REALIGN still may introduce realignment errors. DOCUMENT is used as an additional score to contextualize the performance. More about the realignment methodologies is in Section 2.

5 Segmentation at Training with a Glass Box System

Segmentation occurs at an early stage in the NMT pipeline, so it is intuitive to think it could have a large effect: Incorrect segmentations can lead to incorrect alignments; incorrect alignments lowers the quality of the training data; and low quality training data will produce worse models.

In order to study the effects on training, we recreate the NMT pipeline by segmenting documents and aligning bitext to train models. We apply each segmenter to the training data resulting in a new unique set of “sentences" for each segmenter. We can then align these sentences to create a unique dataset.

5.1 Document-based Alignment

The standard training paradigm for machine translation identifies bilingual document pairs, segments the sentences on both sides, and then aligns. The alignments are ideally one-to-one, but often many-to-one (or one-to-many) alignments are also permitted. The product of this is (ideally) tens of millions of sentence pairs that can be used to train machine translation models.

To replicate this, we take monolingual datasets that we know to contain parallel documents. The document alignment is known and labelled. Using these document alignments, we segment and manually re-align the sentences using a document-based aligner.

The document-based aligner we use is Vecalign (Thompson and Koehn, 2019) which uses LASER[10] (Artetxe and Schwenk, 2019) sentence embeddings to compute alignment and also considers many-to-one or one-to-many alignments. This system is a document aligner because it aligns within document context–considering surrounding sentences for many-to-one (or one-to-many) alignments and also constrains the search to alignments along the diagonal (i.e., sentences aligned to each other should occur within a similar placement within their documents).

The number of sentences recovered from the alignment, as well as the average length of source and target in the resulting dataset is shown in Table 2. The difference in size of the resulting datasets is important to note and likely explains the differences in models.

5.2 Global Search Alignment

Recent releases of WikiMatrix (Schwenk et al., 2021a), CCMatrix (Schwenk et al., 2021b), and CCAligned[11] (El-Kishky et al., 2020) have increased the scaling of bitext mining by doing away with the need for bilingual documents.

These techniques extract sentences from monolingual data and attempt to align them with techniques by combining sentence embeddings and clever search algorithms. In such settings, it stands to reason that proper segmentation might be even

[10] `https://github.com/facebookresearch/LASER`

[11] CCAligned somewhat limits the globalness of the search by aligning pseudo documents based on domains.

	Unaligned	Aligned Bitext					
		Document-Based			Global Search		
	total	sents	toks	avg.	sents	toks	avg.
Original	7.5M	7.3M	183.8M	25.0	5.3M	155.1M	29.4
	8.3M		182.6M	24.9		150.9M	28.6
Always	6.5M	5.2M	177.1M	33.8	4.1M	144.6M	35.5
	5.9M		188.5M	35.9		150.1M	36.8
Ersatz	4.9M	4.5M	181.2M	40.4	3.9M	155.3M	39.8
	5.1M		184.5M	41.1		154.8M	39.7
Moses	4.7M	4.3M	182.2M	41.9	3.7M	155.9M	41.8
	5.2M		181.8M	41.8		153.1M	41.1
Punkt	5.0M	4.5M	181.7M	40.1	3.9M	157.0M	39.9
	5.3M		183.4M	40.5		155.5M	39.5
SpaCy	5.8M	5.4M	183.1M	34.0	4.5M	154.7M	34.0
	6.5M		182.3M	33.9		150.7M	33.1
Pairs	3.7M	3.6M	183.8M	51.5	3.0M	160.1M	53.2
	4.2M		185.2M	51.9		156.7M	52.1

Table 2: Training data sizes before (left, Unaligned) and after (right, Aligned Bitext) segmentation and alignment. For each row, the top number denotes the source (`de`) size while the bottom denotes the target (`en`) size. For each segmentation method, displays the total number of retrieved sentence pairs, the total number of tokens (based on white-space), and the average number of tokens in a sentence.

more important, since all alignments are one-to-one. The scaling potential of this technique allows for massive datasets with billions of aligned sentences to produced in many languages. We use the same toolkit used to produce these datasets which uses LASER embeddings and FAISS indexing for quick retrieval.[12]

5.3 Experimental Details

We train a 32,000 joint unigram subword vocabulary using SentencePiece[13] (Kudo, 2018; Kudo and Richardson, 2018) using the original data. We use a Transformer (Vaswani et al., 2017) architecture with 6 encoder and 6 decoder layers. We train with a batch size of 16k tokens validating at the end of each epoch and stopping if the validation has not improved after 10 validations. We validate on WMT19 test sets (with original segmentations). For a comprehensive list of hyperparameters, please Table 7 in the Appendix.

5.4 Results

The amount of data produced by each segmenter and alignment method varied significantly. Data quantity after segmentation and alignment is displayed in Table 2. Vecalign is fairly consistent in the amount of data aligned—roughly 180M tokens with Always creating the highest variance. Vecalign also produces more data in terms of number of sentences compared to the alternative global search method. The global search also varies more significantly with a 10M token difference between the smallest and largest datasets (excluding Always).[14]

We compute the full cross-product of segmentations at training and inference. Results are reported in Table 3. Once again, we find that within a given model, performance is relatively consistent at inference regardless of segmentation. The exception is the Original row as these inputs are under-segmented. This strong mismatch between training and testing points to hallucinations which are further explored in Section 6.

Generally, we see more variation in model performance based on training data segmentation rather than inference segmentation. One of the best performing models was the model trained on the Original data—made by preserving the original segmentations. The prominent feature of its training data was the prevalence of sub-sentence segmentations. We hypothesize this helped in two ways: 1) it was not reliant on a strong end-of-sentence signal (§ 6) and 2) the true alignments were more likely to exist in the training set. If errors in segmentation make alignment difficult, it is beneficial to have segments that are *guaranteed* to correctly align to something. Because the DGT dataset was transla-

[12] `https://github.com/facebookresearch/LASER/blob/main/source/mine_bitexts.py`

[13] `https://github.com/google/sentencepiece`

[14] The default `mine_bitexts.py` setting was used for LASER. The parameters for Vecalign are listed in Table 8 in the Appendix.

	ORIGINAL		ALWAYS		ERSATZ		MOSES		PUNKT		SPACY		PAIRS	
	Vec.	Global	Vec.	Global	Vec.	Global	Vec.	Global	Vec.	Global	Vec.	Global	Vec.	Global
ORIGINAL	25.6	24.9	24.7	16.4	25.0	9.4	22.1	9.8	25.0	11.6	23.8	8.7	28.2	**28.9**
ALWAYS	31.2	**31.7**	30.8	**31.2**	30.4	**31.3**	30.8	30.9	31.3	31.0	30.2	29.8	30.6	**31.4**
ERSATZ	31.3	**31.9**	30.9	31.3	30.6	**31.4**	31.0	31.1	31.3	31.2	30.4	29.9	30.7	**31.5**
MOSES	31.4	**31.9**	30.9	31.3	30.6	31.4	31.0	**31.1**	31.3	31.2	30.4	29.9	30.8	31.6
PUNKT	31.3	**31.9**	30.9	31.3	30.6	**31.5**	31.0	31.1	31.3	31.2	30.4	29.9	30.7	**31.5**
SPACY	**31.4**	**31.8**	30.9	**31.3**	30.7	**31.4**	30.9	31.1	31.4	31.3	30.8	31.2	30.5	**31.6**

Table 3: German–English (de-en) results. The rows denote the segmenter used at *inference* while the columns denote the segmenter used to create the *training data*. The diagonal, thus, has a matching segmenter for both training and inference. The LASER global search alignment method was used to create bitext. Bold denotes significance (p < 0.05) run by paired bootstrapping with sacreBLEU.

tion memory, most segments had a true alignment.

In the ORIGINAL inference-time setting, models trained with Vecalign-produced bitext performed better than their Global counterparts. We hypothesize this is because Vecalign was able to recover many-to-one or one-to-many alignments where the Global aligner was not. This made the models more robust to many-sentence inputs and outputs.

Lastly, we note that the choice of segmenter *does* affect the training data, and thus the final trained model. The ORIGINAL model often had the highest BLEU score across inference-time segmentations. The differences between the ORIGINAL model, and the ERSATZ and PAIRS models were not statistically significant in most cases. Models trained on data created by PUNKT, SPACY, or MOSES (often used to create MT datasets) were not as competitive.

6 Qualitative Analysis

Hallucinations, or addition of content during translation, and deletions are common in neural machine translation. These models are no exception. Qualitative analysis reveals two types of errors that are worth investigating further: 1) seemingly arbitrary deletion of content when the input is unsegmented 2) addition of content without a true signal in the source. We suspect the explanations for these behaviors are 1) a lack of many-sentence inputs occurring in training data and 2) incorrect segmentations leading to poorly aligned data. We display some examples in Table 4.

6.1 Deletion

Almost all models fail when given unsegmented data at inference (the ORIGINAL row). Upon inspection of these translations, it is obvious the reasons for these scores. In Table 4, we show an instance of this in the first column. The source input has three sentences. Some models trained with segmenters (ERSATZ, MOSES, PUNKT, and SPACY) drop the majority of these sentences. The models trained with the ORIGINAL segmentations and the ALWAYS segmentation method incorporate information across sentences and hallucinate new conjunction methods (inserting “with" or using commas). The model trained with the PAIRS setting does a combination. As this setting often has two sentences per line in the bitext, this translation also is limited to two lines and similar, to PUNKT and ORIGINAL, hallucinates ways to combine these sentences. We can infer the reason for the drop in BLEU scores in the unsegmented setting is because most models are *deleting* content. In order to report the prevalence of deletion, we report how many sentences were deleted during translation.

There are 785 lines in the test data but most of the lines contain more than one sentence. We can use Moses (one of the segmenters that is quite conservative—prone to undersegmenting) to count how many sentences occur in the source input as well as how many sentences occur in the translated outputs. In Table 5, we display this information.

The fact that PAIRS translates more sentences is logical as its training data often had *pairs* of sentences in the training data. The ORIGINAL setting translating more sentences than other models seems counterintuitive as the training data was shorter on average (see Table 2). We suspect that the reason for this is more related to the fact that many training examples in the ORIGINAL setting *did not end in punctuation* since they were below the sentence level. In the ERSATZ training data, for example, 98% of training example's target sequences end with a period. Conversely, 62% of the ORIGINAL data ended with a period. We reason that the model was not highly likely to end the sequence after de-

	DELETION	ADDITION
SOURCE	Für Online-Händler sind viele zurückgeschickte Pakete verlorene Ware. Rund 20 Millionen Retouren landen so auf den Müll. Doch gibt es eine Alternative.	Der Premier droht damit, das Land am 31. [Oktober ohne Abkommen aus der EU zu führen...]
ORIGINAL	For online traders, many returned packages are lost commodities, with around 20 million retours pouring into the rubbish, but there is an alternative.	The Prime Minister is threatening the country on 31 December.
ALWAYS	For online traders, many returned packages are lost, but there is an alternative.	The Prime Minister is threatening to leave the country on 31 December.
ERSATZ	Some 20 million retours thus end up in the rubbish.	The Prime Minister is threatening to hold the country on 31 May.
MOSES	For online traders, many returned packages are lost goods.	The Prime Minister is threatening to do so, the country on 31 December.
PUNKT	For online dealers, many returned packages are lost goods.	The Prime Minister is threatening to see the country on the 31st day of the month.
SPACY	For online traders, many returned packages are lost goods.	The Prime Minister is threatening the country on 31 December.
PAIRS	For online traders, many returned packages are lost goods, with some 20 million retours ending up in rubbish. But there is an alternative.	The Prime Minister is threatening the country on 31 December.

Table 4: Examples of differences in translations. The SOURCE denotes input to the model. Content in square brackets was not part of input but has been included for context to reader. The DELETION column shows examples of different models deleting content during translation due to unsegmented input. The ADDITION column shows models hallucinating content when an incomplete input was given.

MODEL	SENTENCES
REFERENCE	1959
ORIGINAL	1009
ALWAYS	785
ERSATZ	793
MOSES	790
PUNKT	830
SPACY	790
PAIRS	1399

Table 5: Number of sentences (as counted by Moses segmenter) generated on the WMT20 de-en test set. The model is each translation system trained on data segmented by the specified segmenter.

coding a period due to these trends.

All segmenter models were affected by this behavior, but SPACY had more issues. SPACY, in a manner different to the other segmenters, also includes punctuation such as ':' as final punctuation meaning it oversegments in many scenarios. In sentences including colons, we see similar deletion from SPACY. For instance, the SPACY model translates ***“Katastrophe abgewendet: Großbrand in französischem Chemiewerk gelöscht."*** simply as ***“Avoiding disaster:"***

Of the 157 sentences in the test data that included a colon, SPACY translated *only* the first segment in 78 of them; in 52 it translated only the second segment; in 27 it translated parts of both.

6.2 Additions

When Raunak et al. (2021) studied the causes of hallucinations, they attributed hallucinations to errors in bitext alignment. It follows that segmentation, as a precursor to bitext alignment, might also affect hallucinations. The most obvious hallucination we see in the translations is surrounding dates. German often uses a format of “Freitag, 27. September 2019" for “Friday, September 27, 2019". Erroneous segmentation around the punctuation in the date causes alignment issues or bad input to the translation model. We see the effects of both.

The former, bad alignment, we see in the case of the overly-aggressive segmenter (ALWAYS). Because the data was always split on the date in this construction, the alignment suffers severely. We see examples in the training data such as:

Source: Juli 2016 an.
Target: Done at Brussels, 20 July 2016.

The training data frequently contains dates like this and the global search aligner was unable to detect that additional information appeared on the target side. The ALWAYS model memorized this extraneous information and generated it 8 times in these experiments.

In the second case, where the input to the model has been over-segmented, we see a similar effect. When an ALWAYS segmenter is used at inference, the models struggle on the incomplete information. In the second column of Table 4, a complete sentence has been segmented erroneously into two incomplete sentences. For clarity the end of the sentence (which was segmented into a separate input) is included in square brackets. The incomplete input has a two-fold effect: 1) the models hallucinate months to attach to the date 2) ALWAYS, ERSATZ, and PUNKT hallucinate verbs (to leave, to hold, to see respectively).

7 Related Works

To the best of our knowledge, not much work has been done about the effects of segmentation on down stream tasks. Raunak et al. (2021) investigates corpus-level noise and empirically links noise patterns to types of NMT hallucinations. Other work has focused on the effects that punctuation has on neural language models (Ek et al., 2020; Karami et al., 2021). In online simultaneous speech segmentation, Wang et al. (2019) proposes an online sentence segmentation approach which improves downstream BLEU scores.

There is much more work pushing away from the sentence-level paradigm and encouraging document translation. Sun et al. (2022) has recently shown that modern neural architectures still achieve strong performance with longer, multi-sentence inputs. A spate of recent work has gone into better document evaluation metrics (Jiang et al., 2022; Vernikos et al., 2022). Document pairs contain the additional context needed to correctly translate certain discourse phenomenon such as coreference resolution and consistent lexical choices. Further, mining documents instead of sentences circumvents the error propagation from using various segmentation methodologies during bitext mining.

8 Conclusion

An NMT system trained on segmented data requires segmentation at inference; however, the exact method of segmentation at inference seems to have little quantitative effect. The larger impact of segmentation occurs during the creation of bitext. Whether the effect stems from the quality of the produced sentence pairs or the limitations of different alignment methods cannot be determined based on these results. Despite this, various segmentation and alignment method combinations create significantly different amounts of bitext to train models on–something that needs to be investigated further. The differences in the resulting data produce models that perform differently. Lastly, we note that when models are trained on segmented data, they dramatically hallucinate at inference with unsegmented data by deleting long segments. By adding some amount of unsegmented data in the training data, this effect can be mitigated to recover upwards of 4 BLEU points.

Together, we might conclude that avoiding segmentation is the path forward. When the segmentation and alignment techniques failed, half a million sentence pairs were sometimes lost or left unaligned. Additionally, we see that less-segmented bitext produces models that are more robust to unsegmented data at inference. The biggest hurdle in training document-level models is the lack of sufficient document-level annotations. If true document pairs exist in larger web-scraped corpora, most of the original document structure (and informative context) has been removed via bitext filtering and deduplication. Future work might explore potential solutions to mining document-level data, and circumvent these segmentation tools and their respective noise.

9 Limitations

Most of this work relies on interactions between the segmenters and the aligners. It's the production of training data—and the resulting quality and quantity that is causing the differences in models. We used off-the-shelf configurations for these aligners and didn't do significant hyper-parameter searching. It's possible that other toolkits or different hyperparameters might normalize the effects of erroneous segmentation.

We also noted that Vecalign was able to recover erroneous segmentation in the one-to-many and many-to-one settings while showing that the global method was not. Having a global search method does not directly preclude these recoveries, but to the best of our knowledge it hasn't been investigated.

Lastly, we limit ourselves to `de-en` as a language pair here because of the availability of document pairs. The ambiguity in puncutation surrounding these two languages make them interesting for segmentation. Also, German often uses a different word order than English which can make aligning erroneous segmentations difficult. These effects might be minimized or non-existant in other language pairs.

References

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Adam Ek, Jean-Philippe Bernardy, and Stergios Chatzikyriakidis. 2020. How does punctuation affect neural models in natural language inference. In *Proceedings of the Probability and Meaning Conference (PaM 2020)*, pages 109–116, Gothenburg. Association for Computational Linguistics.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5960–5969, Online. Association for Computational Linguistics.

William A. Gale and Kenneth W. Church. 1993. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102.

Yuchen Jiang, Tianyu Liu, Shuming Ma, Dongdong Zhang, Jian Yang, Haoyang Huang, Rico Sennrich, Ryan Cotterell, Mrinmaya Sachan, and Ming Zhou. 2022. BlonDe: An automatic evaluation metric for document-level machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1550–1565, Seattle, United States. Association for Computational Linguistics.

Mansooreh Karami, Ahmadreza Mosallanezhad, Michelle V. Mancenido, and Huan Liu. 2021. "let's eat grandma": When punctuation matters in sentence representation for sentiment analysis. *CoRR*, abs/2101.03029.

Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Robert C. Moore. 2021. Indirectly supervised english sentence break prediction using paragraph break probability estimates. *CoRR*, abs/2109.12023.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The curious case of hallucinations in neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1172–1183, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021a. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021b. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500, Online. Association for Computational Linguistics.

Zewei Sun, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Shujian Huang, Jiajun Chen, and Lei Li. 2022. Rethinking document-level neural machine translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3537–3548, Dublin, Ireland. Association for Computational Linguistics.

Brian Thompson and Philipp Koehn. 2019. Vecalign: Improved sentence alignment in linear time and space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1342–1348, Hong Kong, China. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121, Online. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Giorgos Vernikos, Brian Thompson, Prashant Mathur, and Marcello Federico. 2022. Embarrassingly easy document-level MT metrics: How to convert any pretrained metric into a document-level metric.

Xiaolin Wang, Masao Utiyama, and Eiichiro Sumita. 2019. Online sentence segmentation for simultaneous interpretation using multi-shifted recurrent neural network. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 1–11, Dublin, Ireland. European Association for Machine Translation.

Rachel Wicks and Matt Post. 2021. A unified approach to sentence segmentation of punctuated text in many languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3995–4007, Online. Association for Computational Linguistics.

A Appendix

A.1 Data

In Table 6 is a further breakdown if the amount of sentences extracted from the datasets via each segmenter. In most cases, a segmenter produces more segmentations than the ORIGINAL dataset. This is not true of the DGT dataset which shows how over-segmented it was. Moses is the most conservative segmenter with high-precision and lower recall.

A.2 Hyperparameters

In Table 7, the hyperparameters used to train the Fairseq NMT models are listed. When the parameters are not listed, the defaults were used. Further, we also list the settings used with the Vecalign aligner in Table 8.

	DGT		EUROPARL		NEWS			WMT20
	de	en	de	en	de	en	totals	all
Original	5.24M	6.12M	1.83M	1.83M	0.39M	0.39M	15.80M	12517
Always	4.06M	3.66M	2.06M	1.89M	0.41M	0.39M	12.47M	17434
Ersatz	2.62M	2.88M	1.93M	1.83M	0.40M	0.39M	10.04M	16132
Moses	2.37M	3.05M	1.90M	1.79M	0.39M	0.39M	9.89M	15915
Punkt	2.63M	3.01M	1.97M	1.86M	0.40M	0.38M	10.25M	16137
SpaCy	3.33M	4.15M	2.06M	1.97M	0.43M	0.42M	12.35M	17342
Pairs	2.63M	3.07M	0.92M	0.92M	0.20M	0.20M	7.94M	-

Table 6: Sizes of the source (`de`) and target (`en`) after applying segmentation techniques described in Section 2. These sizes are before alignment. To the right (WMT20), we list the sizes of the segmented test sets (all 12 languages together).

Parameter	Value
Architecture	Transformer
Encoder Layers	6
Decoder Layers	6
Embed Dim	512
FFN Dim	512
Attention Heads	8
Dropout	0.1
Attn. Dropout	0.1
ReLU Dropout	0.1
Label Smoothing	0.1
Adam Betas	(0.9, 0.98)
Clip Norm	2.0
Lr Scheduler	Inverse Sqrt
Warmup Updates	4000
Initial LR	1e-7
LR	0.0005
Min LR	1e-9
Batch Size	16k tok
Patience	10

Table 7: Values for the hyperparameters used during training. Can be traced to the Fairseq parameters. If not listed, default was used.

Parameter	Value
Overlap	6
Max Alignment	4
Embedding Model	LASER (93 langs)

Table 8: Settings used with the Vecalign alignment toolkit.

Revisiting Locality Sensitive Hashing for Vocabulary Selection in Fast Neural Machine Translation

Hieu Hoang[*] **Marcin Junczys-Dowmunt**[*]
Roman Grundkiewicz **Huda Khayrallah**
Microsoft, 1 Microsoft Way, Redmond, WA 98052, USA
{hihoan,marcinjd,rogrundk,hkhayrallah}@microsoft.com

Abstract

Neural machine translation models often contain large target vocabularies. The calculation of logits, softmax and beam search is computationally costly over so many classes. We investigate the use of locality sensitive hashing (LSH) to reduce the number of vocabulary items that must be evaluated and explore the relationship between the hashing algorithm, translation speed and quality. Compared to prior work, our LSH-based solution does not require additional augmentation via word-frequency lists or alignments. We propose a training procedure that produces models, which, when combined with our LSH inference algorithm increase translation speed by up to 87% over the baseline, while maintaining translation quality as measured by BLEU. Apart from just using BLEU, we focus on minimizing search errors compared to the full softmax, a much harsher quality criterion.

1 Introduction

The computation of the output logit, softmax and beam search (the *output layer*) are some of the most compute-intensive tasks in current Neural Machine Translation (NMT) models, often taking the majority of inference time for many models, on many hardware architectures, especially in deployment settings. This is mainly due to the large vocabulary size relative to other dimensions in the model. Methods that reduce the effective vocabulary size can have a major impact on inference speed. Vocabulary selection is one such method.

However, a known downside of vocabulary selection methods is the risk of search errors if the desired output token $\hat{y}_t$ is not a member of the reduced vocabulary $\overline{V}$, forcing the beam search to choose a sub-optimal token. Even when the impact of such search errors on BLEU is minimal, search errors caused by lexical shortlisting degrade human judgements of quality (Domhan et al., 2022).

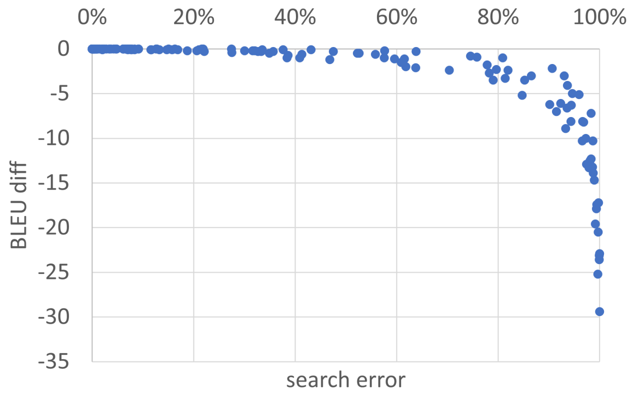


Figure 1: Search errors vs. decrease in BLEU for all experiments in the paper. We vary the two main hyperparameters of our LSH implementation: number of hash functions and the size k of our selected vocabulary subset. 60% of sentences have a search error before we observe a 1 BLEU point degradation.

The degradation in human judgement is less surprising if we inspect Figure 1 which shows that translation quality, as measured by BLEU, is resilient to search errors caused by LSH. Only when over 60% of translations exhibit search errors is there significant BLEU degradation.

We examine vocabulary selection using Locality Sensitive Hashing (LSH), and evaluate specifically in the context of Neural Machine Translation. We introduce an LSH-based vocabulary selection algorithm and compatible models such that:[1]

1. the models have translation quality that is better than or comparable to the baseline model;
2. the LSH-based vocabulary selection algorithm introduces minimal search errors across a number of models and language pairs, including no search errors at all for certain configurations;
3. inference is up to 87% faster than the baseline.

[*] These authors contributed equally to this work.

[1] We release code in Marian; see Section 7.1 for details.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 855 - 869
December 7-8, 2022 ©2022 Association for Computational Linguistics

2 Methods of vocabulary selection

The output layer for a target vocabulary V, performs the following computations:

$$\begin{aligned} p(y_t|y_{1:t-1}, x; \theta) &= \text{softmax}(Wh + b) \\ \hat{y_t} &= \text{argmax}\, p(y_t|y_{1:t-1}, x; \theta), \end{aligned} \tag{1}$$

where $W \in \mathbb{R}^{|V| \times d}$ is the weight matrix, $b \in \mathbb{R}^{|V|}$ is the bias vector, $h \in \mathbb{R}^d$, d is the hidden dimension size of the decoder state, and $p(y_t|y_{1:t-1}, x; \theta) \in \mathbb{R}^{|V|}$ is the softmax probabilities. This is computationally expensive due to the target vocabulary size, $|V|$.

Vocabulary selection create a small subset, $\overline{V} \subset V$. This will reduce the size of weight matrix, $\overline{W}$, and bias vector, $\overline{b}$, where $|\overline{V}| \ll |V|$, $\overline{W} \in \mathbb{R}^{|\overline{V}| \times d}$ and $\overline{b} \in \mathbb{R}^{|\overline{V}|}$.

Equation 1 is then replaced with the more efficient Equation 2 which uses $\overline{W}$ and $\overline{b}$ instead.

$$\begin{aligned} \overline{p}(y_t|y_{1:t-1}, x; \theta) &= \text{softmax}(\overline{W}h + \overline{b}) \\ \overline{\hat{y_t}} &= \text{argmax}\, \overline{p}(y_t|y_{1:t-1}, x; \theta), \end{aligned} \tag{2}$$

The aim is now to find the subset, $\overline{V}$, such that $\overline{\hat{y_t}} = \hat{y_t}$.

Depending on the method, vocabulary selection (and therefore construction of $\overline{V}$, $\overline{W}$ and $\overline{b}$) can be static or happen dynamically per sentence (or batch), per decoder time step, or even per individual decoder hypothesis.

We restrict our overview of the concept of vocabulary selection to the case where the original softmax layer remains largely unmodified except for sub-selection. Methods that require complex structural reformulations of the softmax layer during training like hierarchical softmax (Morin and Bengio, 2005), adaptive softmax (Grave et al., 2016) or binary code prediction (Oda et al., 2017) are outside the scope of this work.

In-depth overviews of past and current vocabulary selection methods are provided by L'Hostis et al. (2016), Shi and Knight (2017), and more recently Domhan et al. (2022). We only repeat concepts that are either common or required to differentiate our work from previous approaches.

2.1 Word frequency-based methods

For simplicity's sake, when describing word frequency-based methods, we assume that vocabulary identifiers correspond to frequency rank (according to a training corpus or other reference corpus) and hence the top-K first items in a vocabulary list are the top-K most frequent words/segments from the training corpus. The choice of K determines a static subset V_f of V where $|V_f| = K$. Then $\overline{V} = V_f$ and the parameters $\overline{W}, \overline{b}$ of the softmax output layer are sub-selected accordingly.

Word frequency-based vocabulary selection is not a viable method on its own — the quality degradation is simply too large to be acceptable (Shi and Knight, 2017) — but it constitutes an important common back-bone for several of the more accurate methods discussed below as it is an easy way to include common segments like function words, punctuation, etc. in the output vocabulary.

2.2 Word alignment-based methods

Word alignment-based vocabulary selection (Jean et al., 2015a) has been part of the NMT toolbox since the earliest competitive NMT systems. Jean et al. (2015b) first introduce the concept in essentially the form it is widely being used today[2] in their submission to the WMT15 shared task (Bojar et al., 2015). Later work (Mi et al., 2016; L'Hostis et al., 2016; Shi and Knight, 2017) rediscover mostly the same setup or confirm it to be one of the strongest methods amongst a number of other approaches.

Given a source sentence $x_{1:m}$ and a word-alignment dictionary with alignment probabilities between source and target segments $p_a(y|x)$, this method creates $V_a = \bigcup_{t \in 1:m} V_a(x_t)$, where for instance $V_a(x_t) = \{y \in V : p_a(y|x_t) \geq \overline{p}\}$ for a given threshold $\overline{p}$. Other criteria for constructing $V_a(x_t)$ are possible: such as K' most probable aligned target words or combinations of multiple criteria.

Finally, the alignment-based method is typically combined with the frequency-based method as $\overline{V} = V_f \cup V_a$. Word alignment thus extends and refines the target word-frequency method by mapping source sentence context to plausible target language vocabulary candidates (ranked or selected by translation probability). Note, that V_a is constructed dynamically once per source sentence or batch which forces a dynamic construction of $\overline{V}$.

2.3 Earlier LSH-based approaches

Locality Sensitive Hashing (LSH) as a way to accelerate the computation of inner products has been

[2] See submissions to the recent shared tasks on efficient NMT (Hayashi et al., 2019; Heafield et al., 2020, 2021).

investigated as early as 2014 (Vijayanarasimhan et al.) for handling large vocabularies and remains an active area of research for more general neural network training (see e.g. Chen et al., 2020).

Previous work on using LSH in NMT (Shi and Knight, 2017; Shi et al., 2018) takes an approach that is analogous to the alignment-based methods in the sense that a static vocabulary based on word frequency is extended with target vocabulary items that are plausible in the dynamic context of the decoded sentence. However, instead of mapping source segments to target segments via alignment dictionary look-up, the decoder hypothesis state vector h is used to find set $V_l(h)$ of the k target vocabulary items y with the corresponding output layer embedding vector w_y most *similar* to h. As before, the static word-frequency based vocabulary set is merged with the contextual set to form $\overline{V}(h) = V_f \cup V_l(h)$. Note however, that $\overline{V}$ now depends dynamically on each decoder state h.

The specifics of how similarity between the vectors is defined determine the speed and accuracy of the method. The output layer itself can be seen as a similarity function (inner product with softmax normalization) that has perfect accuracy but is least interesting in terms of speed.

Shi and Knight (2017) and Shi et al. (2018) use Winner-Take-All (WTA; Yagnik et al., 2011) hashing with banding to approximate the output layer. However, the type of similarity as expressed via WTA hashing seems to result in fairly low accuracy and therefore needs to be merged with several thousand most frequent vocabulary items to remain competitive in terms of translation quality compared to the full vocabulary.

2.4 Selection as binary classification

L'Hostis et al. (2016) and more recently Domhan et al. (2022) propose to approach the vocabulary selection problem as a per target vocabulary item binary classification problem where each of $|V|$ binary classifiers decides if the corresponding target vocabulary item should be included in the sentence-level (or batch-level) target vocabulary.

L'Hostis et al. (2016) train a suite of $|V|$ binary SVM classifiers which are learned independently from the neural model. The set of words in the source sentence serves as a sparse bag-of-words feature set.

Domhan et al. (2022) train their "neural vocabulary selection" model jointly with the translation model via a multi-objective cost function. They construct $z = \sigma(\text{maxpool}(WH + b))$ where $H \in \mathbb{R}^{d \times m}$ is the hidden encoder context, $W \in \mathbb{R}^{|V| \times d}$, $b \in \mathbb{R}^{|V|}$ and $z \in \mathbb{R}^{|V|}$.

Generally, for both methods, given the binary classifier z_y corresponding to the vocabulary entry y, we have $\overline{V} = \{y \in V : z_y(x_{1:m}) \geq \lambda\}$ where λ is the decision threshold for including y in $\overline{V}$. $\overline{V}$ is constructed dynamically once per sentence and both methods do not need to be merged with the word-frequency-based vocabulary list V_f. The threshold λ seems to be sufficient to control for speed versus accuracy trade-offs.

3 Our LSH-based method

Our work contrasts with prior research on LSH for NMT by Shi and Knight (2017); Shi et al. (2018) in that:

1. We use SimHash hash instead of WTA hash.
2. We do not need to expand the LSH vocabulary subset $\overline{V}$ by merging with a static list of the most frequent words.
3. We do not need to merge $\overline{V}$ across batch and beam entries.
4. We create $\overline{V}$ by finding the top-k smallest Hamming distances, rather than banding hashes and Cuckoo lookups.
5. Our target vocabulary is smaller than most experiments in the above works which experimented with target vocabulary sizes of 66k, 50k, 40k and 25k. We believe larger vocabularies are unnecessary as a result of the use of sub-word units (Sennrich et al., 2016) and their variants. We use sub-word units while the above works do not.
6. We are concerned with search errors introduced by vocabulary selection as well as with translation quality degradation. Quality metrics are often insensitive to errors caused by deviation from an otherwise unfiltered vocabulary.

3.1 SimHash for Softmax approximation

Prior research on the application of LSH for NMT by Shi and Knight (2017); Shi et al. (2018) relies on WTA hashing. We found SimHash (Charikar, 2002) to result in much lower search error.

For a random normal vector $r \in \mathbb{R}^d$ and an input vector v (of the same size as r), SimHash introduces the following hash function H_r:

$$\mathrm{H}_r(v) = \begin{cases} 1 & \text{if } v \cdot r \geq 0 \\ 0 & \text{if } v \cdot r < 0 \end{cases}$$

which maps v to a single bit. The above is generalized to c bits by generating and applying c different random vectors and concatenating the results. This can be simplified via multiplying with a projection matrix $R \in \mathbb{R}^{d \times c}$ and the same dimension-wise mapping to bits of the result.[3] We call this function $\mathrm{H}_R(v) : \mathbb{R}^d \rightarrow \{0, 1\}^c$ and use it to obtain the LSH representation of v. Further, $\mathrm{D}(\mathrm{H}_R(u), \mathrm{H}_R(v))$ denotes the bit-wise Hamming distance between the hashed binary representations of vectors u, v.

SimHash has been designed in such a way that for two vectors u, v for which the angle $\theta(u, v)$ between these vectors is small, the Hamming distance D over their hashed binary vectors should be small as well.[4] Naturally, the cosine similarity $\cos(\theta(u, v))$ will be high for such cases.

It is this property which allows us to apply a series of transformations and approximations to find a promising candidate for the most probable vocabulary item $\hat{i}$ for a decoder state vector h (and the output layer parameters W and b) using fast Hamming distance computation:

$$\begin{aligned} \hat{i} &= \operatorname*{argmax}_{i \in V} \operatorname*{softmax}_{i}(Wh + b) && (3) \\ &= \operatorname*{argmax}_{i \in V} w_i \cdot h + b_i && (4) \\ &\approx \operatorname*{argmax}_{i \in V} w_i \cdot h && (5) \\ &\approx \operatorname*{argmax}_{i \in V} \cos(\theta(w_i, h)) && (6) \\ &\approx \operatorname*{argmax}_{i \in V} \cos\left(\mathrm{D}(\mathrm{H}_R(w_i), \mathrm{H}_R(h)) \frac{\pi}{c}\right) && (7) \\ &= \operatorname*{argmin}_{i \in V} \mathrm{D}(\mathrm{H}_R(w_i), \mathrm{H}_R(h)). && (8) \end{aligned}$$

In every step above which leads with $\approx$, we introduce a new approximation to the previous step, potentially reducing the accuracy of the search for $\hat{i}$. When moving from Equation 4 to Equation 5, we drop the bias term b_i as it cannot be easily incorporated in the search in Hamming space. For models with large values in the bias vector b, this will inadvertently lead to search errors. The easy solution to this problem is to drop the bias term during training as well. More on this in Section 5.1.

In Equation 6 we ignore the magnitude of the vectors. This seems to not matter much for the search and we leave investigating the effects or potential mitigation for future work.[5]

Equation 7 sees the introduction of the SimHash LSH as we approximate the angle θ via the Hamming distance. Finally, in Equation 8 we can find the most promising vocabulary candidate by directly minimizing the Hamming distance; note that we flipped from argmax to argmin.

3.2 Integrating LSH with beam search

In Section 2, we categorized methods of vocabulary selection by how and when they construct the set of subselected vocabulary $\overline{V}$.

Before translation begins, the output embedding weights W are hashed once using the SimHash function $\mathrm{H}_R(W)$ to create a set $L \in \{0, 1\}^{|V| \times c}$ of LSH keys, one for each target vocabulary entry from V:

$$L = \{l_1, \ldots, l_{|V|}\} = \mathrm{H}_R(W). \qquad (9)$$

Similar to the other LSH-based methods from Section 2.3, we construct $\overline{V}(h)$ dynamically for every decoder state h. During each decoding step the same hash function is applied to the decoder state h to obtain a hashed binary query $q \in \{0, 1\}^c$:

$$q = \mathrm{H}_R(h).$$

If the transformations in Equation 3 to Equation 8 were exact, we would only need to find the vocabulary candidate i corresponding to key $l_i \in L$ with the lowest Hamming distance from the query q (in the case of greedy decoding). However, all the approximations lead to search errors and we investigate a set of k best scoring candidates. This set

[3] Following the LSH implementation in FAISS, we use a Gaussian random rotation matrix $R \in \mathbb{R}^{d \times c}$. If $c \geq d$, FAISS constructs a matrix $R \in \mathbb{R}^{c \times c}$ composed of c orthonormal column vectors via QR factorization and then drops rows until we have $R \in \mathbb{R}^{d \times c}$.

[4] See Charikar (2002) for details. In short, the probability that the hash values for two vectors u, v match is given as $Pr(\mathrm{H}_r(u) = \mathrm{H}_r(v)) = 1 - \frac{\theta(u,v)}{\pi}$. When hashing to bit vectors of length c, the Hamming distance between these bit vectors $\mathrm{D}(\mathrm{H}_R(u), \mathrm{H}_R(v))$ approximates $\frac{\theta(u,v)}{\pi} c$.

[5] The magnitudes of decoder states and weight vectors probably do not vary a lot. However, decoder states h would be normalized to norm $\sqrt{d}$ via layer normalization at no additional computational cost if we dropped the affine transformation after layer normalization. The weight vectors of the output layer could be normalized to unit length after each parameter update during training or via weight normalization (Salimans and Kingma, 2016).

Dataset	de-en	*#sentences* fr-en	es-en
Europarl (Train)	1,920,209	2,007,723	1,965,734
dev2006 (Dev)	2,000	2,000	2,000
nc-dev2007 (Test)	1057	1,057	1,057

Table 1: Data used for training, validation and testing.

is our per decoder state vocabulary subset $\overline{V}(h)$:

$$\overline{V}(h) = \operatorname*{argmin}_{\substack{V' \subset V \\ |V'|=k}} \sum_{i \in V'} \mathrm{D}(q, l_i).$$

In cases where there are more than k elements that would qualify based on their Hamming distance, we retrieve only the first k found. Note, this concludes our construction of $\overline{V}$ and unlike Shi et al. (2018), we do not need to extend $\overline{V}$ with a large word-frequency list.

Next, for every decoder state h, $\overline{W} \in \mathbb{R}^{|\overline{V}| \times d}$ and $\overline{b} \in \mathbb{R}^{|\overline{V}|}$ are subselected from the output layer parameters W and b, respectively, by restricting entries to those corresponding to vocabulary indices in $\overline{V}$. $\overline{W}$ and $\overline{b}$ replace W and b in calculating softmax and the best output token, $\hat{y}_t$, replacing the standard output layer computation in Equation 1 with Equation 2.

The number of hashes per input vector c and the number of target vocabulary to keep k are hyperparameters in our LSH implementation.

4 Experimental Setup

We train a model with 6-layer Transformer (Vaswani et al., 2017) encoder with 6-layer SSRU (Kim et al., 2019) decoder, trained using Marian (Junczys-Dowmunt et al., 2018) and using the same toolkit for inference. This is a strong and realistic model for production MT environments which balances translation quality and efficiency.

We use SentencePiece (Kudo and Richardson, 2018) with 32,000 tokens for all models, shared between both source and target language.

We use the FAISS (Johnson et al., 2019) implementation of SimHash hash described in Section 3.1.

We trained with four translation directions (German-English, English-German, Spanish-English, and French-English) Europarl corpus (Koehn, 2005), validated on the held out development set from the same corpus (*'dev2006'*) and tested on the out-of-domain New Commentary test set (*'nc-dev2007'*). See Table 1 for data set sizes. Results are reported for German-English in Section 5, results for other language pairs are available in the Appendix 7.2.

We use a two stage training procedure. In the first stage, we train a translation model directly on the parallel data. We create a synthetic parallel corpus by translating the source side of the parallel corpus with the initial model. The original source is paired with these translations to form the synthetic corpus for stage two. For the second stage of training, we then consider two cases: (1) where the 2nd stage model topology is identical (i.e. the original model and the new model both have or lack the output bias) and (2) where there is some change in model topology.

For case (1), we use *self-training*: the first-stage model are fine-tuned using the synthetic corpus. For case (2): we use sequence-level *knowledge distillation*: new models are distilled (Hinton et al., 2015) by training from scratch on the synthetic corpus. By abuse of terminology, in both scenarios, we call the first-stage model the *teacher*, and the fine-tuned or distilled model the *student*.

Translation quality was measured using SacreBLEU (Post, 2018). We define *search errors* as the number of lines changed in the translation output when vocabulary selection is applied.

We measure the time taken to do inference on one core of a 12 core Intel Xeon CPU, on a PC with 16GB RAM, running Ubuntu 20.04 within a WSL2 hypervisor.

For short listing, we create a candidate list of target sub-word translations for each source sub-word by using word alignments obtained from FastAlign (Dyer et al., 2013). A shortlist of target sub-words is created before the translation of each sentence to constrain the possible output sub-words.

5 Results and analysis

Table 2 shows results on the full teacher-student training procedure, compared to the baseline, teacher, and lexical shortlisting. Our proposed method maintains the same translation quality as greedy search, with a 57% to 80% speedup. By contrast, shortlisting has search errors in 12% to 25% of sentences, with a speedup of between 67% to 74%.

	de-en			fr-en			es-en		
	speed ↑	BLEU ↑	search error ↓	speed ↑	BLEU ↑	search error ↓	speed ↑	BLEU ↑	search error ↓
Teacher	2.84	28.9		2.73	31.0		2.72	40.5	
Student	2.71 (-5%)	**29.9**		2.91 (+7%)	**31.9**		3.01 (+11%)	**42.2**	
Student w/ shortlist (baseline)	**4.76** (+68%)	29.6	25%	4.69 (+72%)	31.8	12%	4.73 (+74%)	42.0	17%
Student w/ LSH (this work)	4.46 (+57%)	**29.9**	**0%**	**4.92** (+80%)	**31.9**	**0%**	**5.08** (+87%)	**42.2**	**0%**

Table 2: Translation speed (sent./sec.), quality (BLEU) and search error for the teacher model, student model with full vocabulary, student model with the shortlist, and student model with LSH vocabulary selection. Hyperparameters chosen for lowest possible search error. Models trained with no bias and with label smoothing, and use a beam size of 1.

	Baseline	Using LSH	
Model	BLEU ↑	BLEU ↑	search error ↓
With bias	28.9	0.1	100%
With bias LS	**29.7**	0.0	100%
No bias	29.1	29.1	7%
No bias LS	29.2	29.2	6%

Table 3: Translation quality (BLEU) for baseline teacher models, and when using LSH ($k = 1024$, $c = 2048$).

In order to understand the contributions of different aspects of the method, we perform additional experiments for analysis.

5.1 The effect of output bias

While the softmax of Equation 1 is dependent on the output bias b as well as output weights W, there is no easy way to include the bias b in the hashed representation of L in Equation 9. To see what effect this omission by the LSH hashing function has on translation, we will train and evaluate models with and without the output bias.

The first column in Table 3 compares the translation quality between models with and without output bias, based on BLEU scores. Models with output bias and training with label smoothing (LS) of 0.1 improve translation quality.

Column two and three in Table 3 show the consequences of applying LSH with the output vocabulary size of $k = 1024$ and the number of hashes set to $c = 2048$ to the baseline models. LSH causes overwhelming search errors in models with output biases, leading to catastrophic collapse in BLEU. This is unsurprising as the LSH does not take the bias into account when computing similarity. On the other hand, models without output bias are not hugely affected by LSH. Between 2% to 7% of the translations suffer from search errors but this has a negligible affect on translation quality.

Model	Beam 1	Beam 4
Teacher with bias	28.9	29.9
+ Student no bias	29.1	29.9
+ Student no bias LS	29.6	30.4
Teacher with bias LS	29.7	30.3
+ Student no bias	28.7	29.3
+ Student no bias LS	29.6	30.2
Teacher no bias	29.1	29.8
+ Student no bias	30.1	**30.9**
+ Student no bias LS	**30.2**	30.8
Teacher no bias LS	29.2	29.9
+ Student no bias	30.0	30.4
+ Student no bias LS	29.9	30.5

Table 4: Translation quality of distilled models on held-out test set (BLEU) with different beam widths.

Based on these results, further experiments with LSH only use models without output bias.

5.2 Comparison with lexical shortlisting

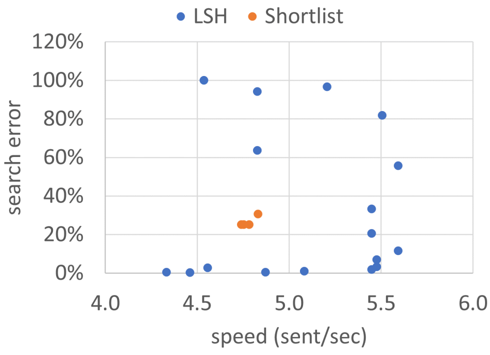


Figure 2: Comparison of translation speed (sent./sec.) vs search error between LSH and lexical shortlisting.

	Beam 1			Beam 4		
	speed ↑	BLEU ↑	search error ↓	speed ↑	BLEU ↑	search error ↓
Teacher model	2.84	29.2		1.52	29.9	
Student model	2.71 (-5%)	29.9		1.47 (-3%)	30.5	
LSH	3.94 (+39%)	29.9	0%	2.01 (+32%)	30.5	13%

Table 5: Translation speed (sent./sec.), quality (BLEU) and search error for student model (trained and distilled with label smoothing, without bias). Compared with teacher without bias or label smoothing. LSH is using parameters 1024-best vocab items, and a hash size of 2048.

Figure 2 shows the translation speed / search error trade-off for LSH and lexical shortlisting for various hyperparameter settings. For shortlisting, We experimented with $|V_f| = 100$ and $|V_a| = 10, 25, 50, 75$ and 100. We have not observed significant changes for larger $|V_a|$. The methods were applied to a model trained, then distilled with no output bias and with layer normalization. Our training procedure and LSH inference algorithm is not only faster than shortlisting but also result in less search errors.

5.3 LSH in teacher-student training

The Hamming distance of the hash vectors in Equation 7 is used in an approximate similarity measure between the decoder state h and each embedding vector corresponding to vocabulary items in V. We would like to increase this similarity for the correct output and decrease it for incorrect output at each time step. Kim and Rush (2016) demonstrated that knowledge distillation create student models with a more peaked distribution, i.e. the probability mass is concentrated around only few vocabulary words. This likely carries over into the space of Hamming distances, separating similar vector pairs from dissimilar ones, a potentially useful phenomenon that the search can take advantage of. See also Section 5.6 for similar considerations on the effects of label smoothing.

Figure 3 shows the trade-off between the LSH top-k versus search errors, for teacher models without and with output bias, and student models without bias, respectively. Similarly, Figure 4 shows the trade-off with the number of hashes c used in LSH. These plots also show that:

1. teacher-student training significantly reduces LSH search errors,
2. teacher models *with* label smoothing have lower search errors,
3. student models *without* label smoothing have lower search errors,
4. all student models converges to minimal, or even zero, search errors with increased top-k.

5.4 Translation speed vs hash size

Predictably, translation speed increases if the LSH parameters decreases. For example, Figure 5 shows the translation speed when the number of hashes are varied. At very low hash counts, the systems often output lengthy sentences with repetitive gibberish, lowering speed. Of course, this has a negative impact on search errors and translation quality.

5.5 Larger beam size

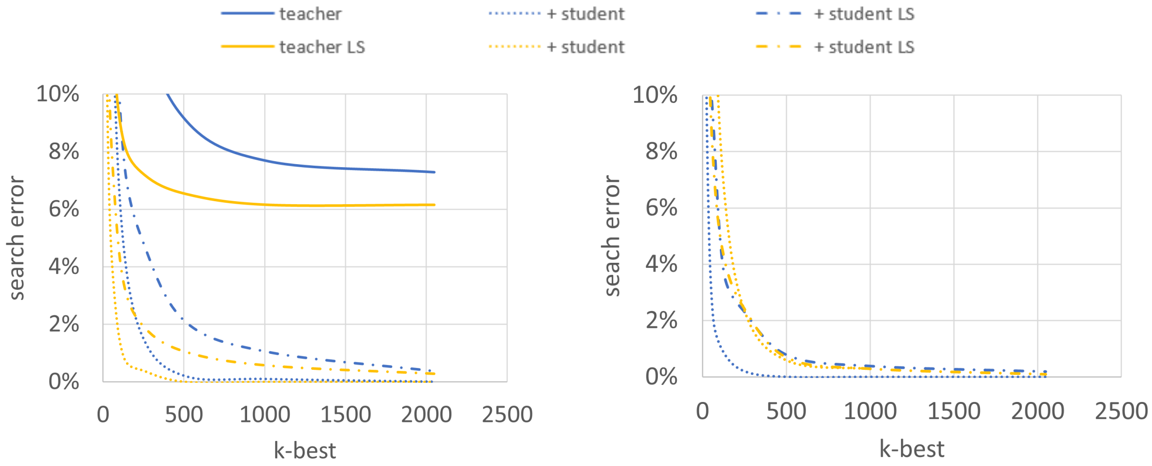
Table 5 shows translation quality when using a larger beam which, as expected, is higher in all cases than using beam width 1.

However, LSH vocabulary selection causes more search errors for larger beam sizes. Figure 6 compare the search errors for the same model using beam width 1 and 4 by varying the LSH hyperparameters. The same conclusion can be drawn from Table 5 (Beam 4).

A possible cause for this increased search error is in the calculation of the softmax denominator. The denominator for each beam is the sum of logits in the beam. When using vocabulary selection, the denominator is approximated by calculating it only over a k-best subset of the logits. The softmax probability would be distorted if the excluded logits contain significant probability mass.

This is not an issue with beam size 1 as an approximation error in the denominator would change the absolute probabilities but won't affect the relative probabilities within the beam.

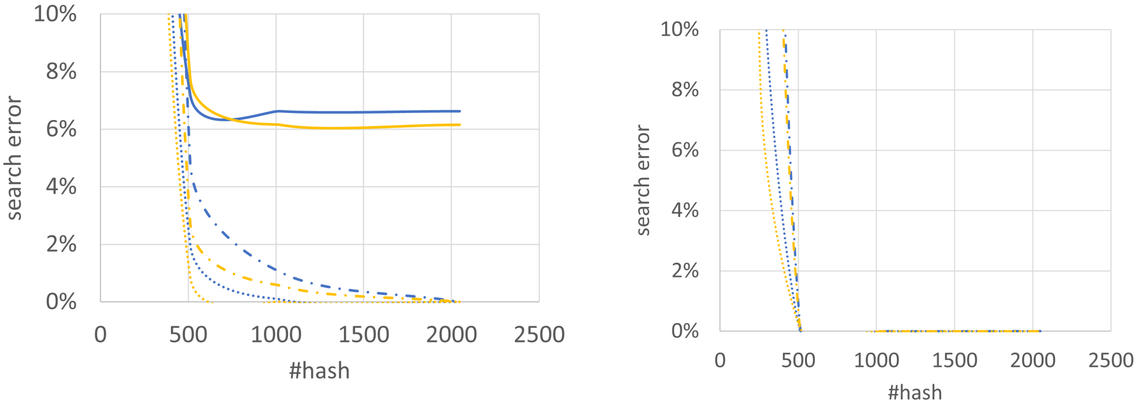
However, for beam size larger than one, probabilities across different beams are compared. A logit approximation error in this case would distort the



(a) without output bias

(b) with output bias

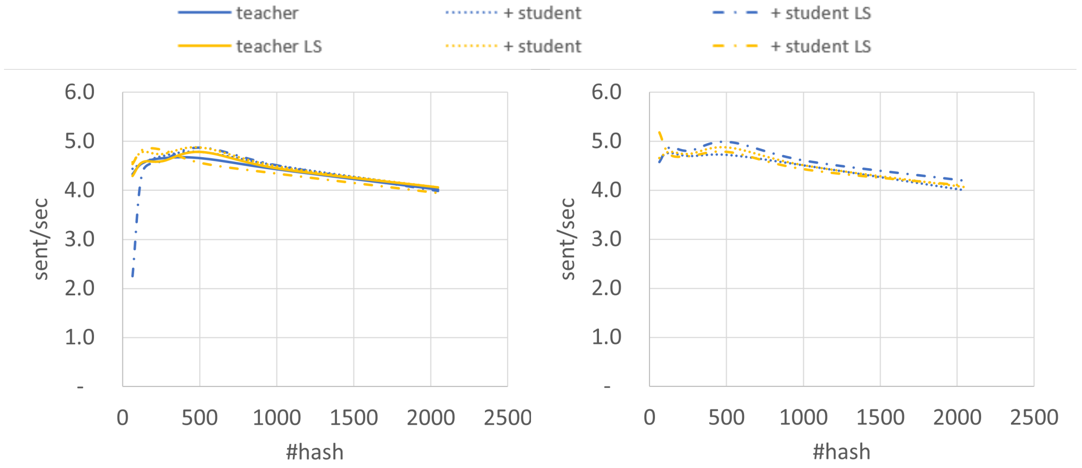
Figure 3: LSH k-best vs. search errors.



(a) without output bias

(b) with output bias

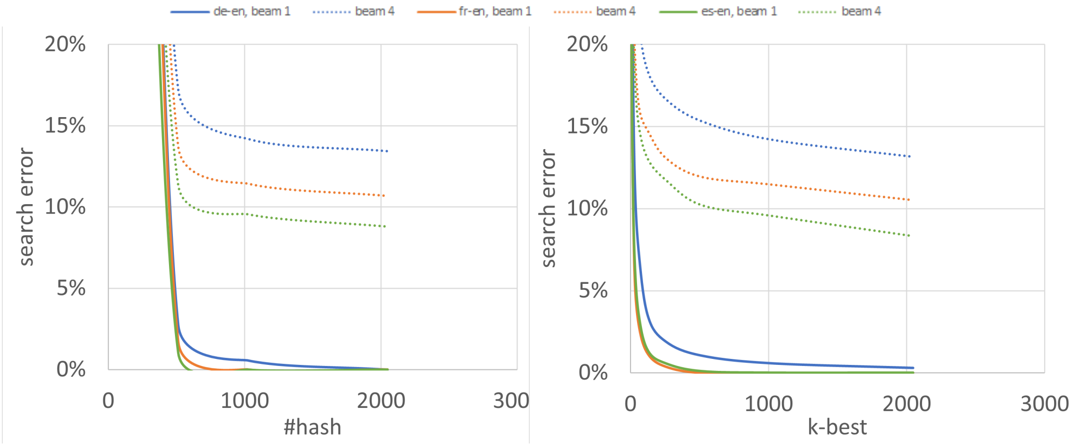
Figure 4: LSH #hashes vs. search errors.



(a) Without bias

(b) With bias

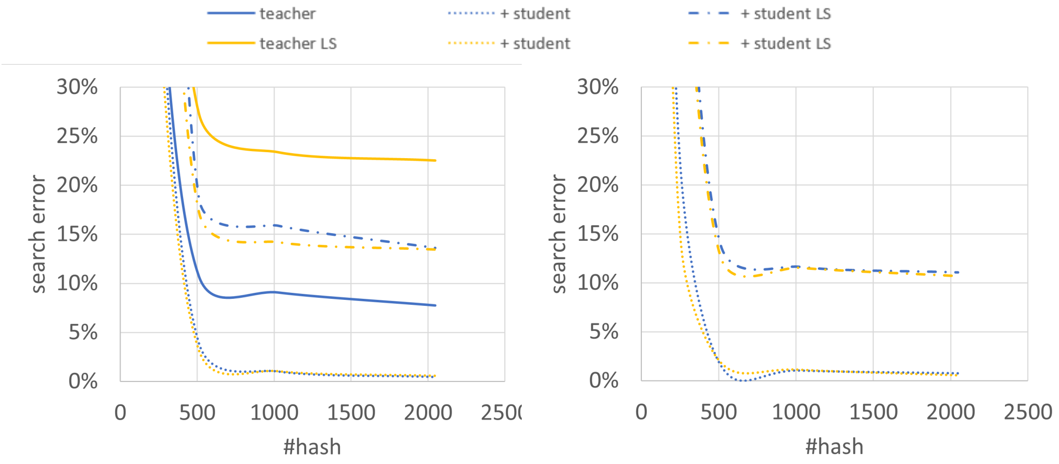
Figure 5: Translation speed (sent./sec.) vs #hashes



(a) Search error vs. #hashes

(b) Search error vs. LSH k-best

Figure 6: Search error for beam size 1 & 4: teacher LS + student LS.



(a) teacher without output bias

(b) teacher with output bias

Figure 7: LSH #hashes vs. search errors for teacher models, beam width 4.

comparison between vocabulary items in different beam, leading to search errors.

5.6 Label smoothing

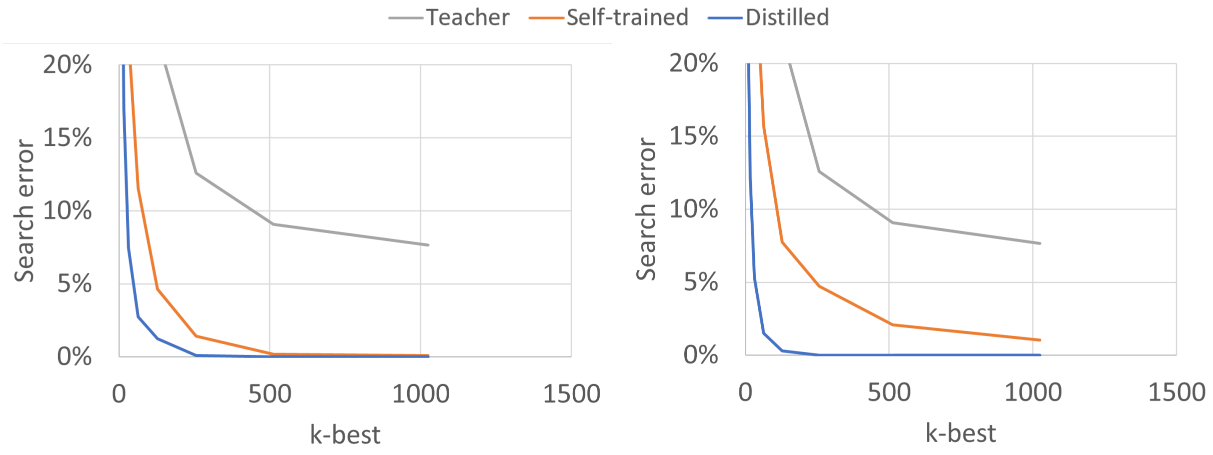
While label smoothing (Szegedy et al., 2016) can improve translation quality—by spreading the probability over many output classes to avoid overfitting— models with label smoothing are detrimentally affected by larger beam widths when LSH is used. Figure 7 shows that both student and teacher models have higher search errors when trained with label smoothing. Since label smoothing distributes a portion of the probability mass over the entire vocabulary, the excluded logits will contain a larger amount of the total probability mass, exacerbating the problem caused by the larger beam size. Since label smoothing may also reduce information transfer in knowledge distillation (Müller et al., 2019), we recommend training students without label smoothing, especially when using larger beams.

5.7 Self-training vs distillation

Thus far, we have fined-tuned ('self-trained') models where the second stage model is architecturally identical to the first, otherwise we distilled a student model from the first stage model.

For first stage models with no bias, Table 6 shows that fine-tuning result in better translation quality than training from scratch with the synthetic data.

However, the fine-tuned models have slightly higher search errors, nevertheless both training strategies result in models which have much lower search errors than the original first stage model, Figure 8.



(a) no label smoothing

(b) label smoothing 0.1

Figure 8: LSH Fine-tuned vs. distilled models. Both are teachers & student with no bias

Model	Beam 1	Beam 4
Teacher no bias	29.1	29.8
+ Self-trained no bias	30.1	**30.9**
+ Self-trained no bias LS	**30.2**	30.8
+ Distilled no bias	29.3	29.9
+ Distilled no bias LS	29.5	30.5
Teacher no bias LS	29.2	29.9
+ Self-trained no bias	30.0	30.4
+ Self-trained no bias LS	29.9	30.5
+ Distilled no bias	29.5	30.3
+ Distilled no bias LS	29.8	30.3

Table 6: Translation quality of fine-tuned vs. distilled models (BLEU).

6 Conclusion

We demonstrate that, with the proper training procedure, using locality sensitive hashing for vocabulary selection can significantly boost translation speed while consistently producing negligible search errors.

We make the following recommendations for use in practice:

For existing models and greedy search, perhaps where we may not know the exact training procedure and model, we can create a model that works with LSH vocabulary selection by distilling the original model to a comparable model without output bias. Using label smoothing in the distillation can improve its translation quality if the original model was not trained with it. There will be minimal search errors in using LSH while achieving significant speed improvement.

To train a new model for use with greedy search, a two stage procedure should also be used where the second stage is fine-tuned on the output of the first. Both stages should train models without output bias. Again, the fine-tuned models can be trained with label smoothing without affecting the effectiveness of LSH.

LSH vocabulary selection introduce search errors for larger beam sizes, especially when label smoothing is used during fine-tuning. Therefore, if using larger beams in inference, it is recommended not to use label smoothing in the distillation or fine-tuning step.

References

Ondřej Bojar, Rajan Chatterjee, Christian Federmann, Barry Haddow, Chris Hokamp, Matthias Huck, Varvara Logacheva, and Pavel Pecina, editors. 2015. *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Lisbon, Portugal.

Moses Charikar. 2002. Similarity estimation techniques from rounding algorithms. In *STOC '02*.

Beidi Chen, Tharun Medini, James Farwell, Sameh Gobriel, Charlie Tai, and Anshumali Shrivastava. 2020. Slide : In defense of smart algorithms over hardware acceleration for large-scale deep learning systems.

Tobias Domhan, Eva Hasler, Ke Tran, Sony Trenous, Bill Byrne, and Felix Hieber. 2022. The devil is in the details: On the pitfalls of vocabulary selection in neural machine translation. In *NAACL*.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Edouard Grave, Armand Joulin, Moustapha Cissé, David Grangier, and Hervé Jégou. 2016. Efficient softmax approximation for gpus.

Hiroaki Hayashi, Yusuke Oda, Alexandra Birch, Ioannis Konstas, Andrew Finch, Minh-Thang Luong, Graham Neubig, and Katsuhito Sudoh. 2019. Findings of the third workshop on neural generation and translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 1–14, Hong Kong. Association for Computational Linguistics.

Kenneth Heafield, Hiroaki Hayashi, Yusuke Oda, Ioannis Konstas, Andrew Finch, Graham Neubig, Xian Li, and Alexandra Birch. 2020. Findings of the fourth workshop on neural generation and translation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 1–9, Online. Association for Computational Linguistics.

Kenneth Heafield, Qianqian Zhu, and Roman Grundkiewicz. 2021. Findings of the WMT 2021 shared task on efficient translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 639–651, Online. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015a. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Beijing, China. Association for Computational Linguistics.

Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015b. Montreal neural machine translation systems for WMT'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 134–140, Lisbon, Portugal. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Young Jin Kim, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019. From research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 280–288, Hong Kong. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71. Association for Computational Linguistics.

Gurvan L'Hostis, David Grangier, and Michael Auli. 2016. Vocabulary selection strategies for neural machine translation. *CoRR*, abs/1610.00072.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Vocabulary manipulation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 124–129, Berlin, Germany. Association for Computational Linguistics.

Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In *AISTATS*.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Yusuke Oda, Philip Arthur, Graham Neubig, Koichiro Yoshino, and Satoshi Nakamura. 2017. Neural machine translation via binary code prediction.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Tim Salimans and Diederik P. Kingma. 2016. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *NIPS*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Xing Shi and Kevin Knight. 2017. Speeding up neural machine translation decoding by shrinking run-time vocabulary. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 574–579, Vancouver, Canada. Association for Computational Linguistics.

Xing Shi, Shizhen Xu, and Kevin Knight. 2018. Fast locality sensitive hashing for beam search on gpu.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Sudheendra Vijayanarasimhan, Jonathon Shlens, Rajat Monga, and Jay Yagnik. 2014. Deep networks with large output spaces.

Jay Yagnik, Dennis Strelow, David A. Ross, and Ruei-sung Lin. 2011. The power of comparative reasoning. In *2011 International Conference on Computer Vision*, pages 2431–2438.

7 Appendix

7.1 Practical Considerations

Here, we discuss some practical considerations for use of the LSH decoding.

To train a Marian model without an output bias, add the following switches.
Marian training command:[6]

```
--output-omit-bias
```

To train without label smoothing:

```
--label-smoothing 0
```

To use the LSH vocabulary selection during inference, execute marian-decoder with the following switches:

```
--output-approx-knn [k] [c]
```

where [k] is the number of k-best vocabulary items and [c] is the number of hashes to use.

7.2 Results for Additional Language Pairs

Model	de-en	fr-en	es-en
With bias	28.9	31.0	40.5
With bias LS	**29.7**	**31.4**	**41.3**
No bias	29.1	31.3	41.0
No bias LS	29.2	31.2	41.1

Table 7: Baseline translation quality (BLEU) w/ & w/o bias and w/ or w/o label smoothing.

Model	de-en	fr-en	es-en
With bias -	0.1 100%	0.1 100%	0.0 100%
With bias LS -	0.0 100%	0.0 100%	0.0 100%
No bias -	29.1 7%	31.2 4%	41.0 2%
No bias LS -	29.2 6%	31.1 4%	41.1 2%

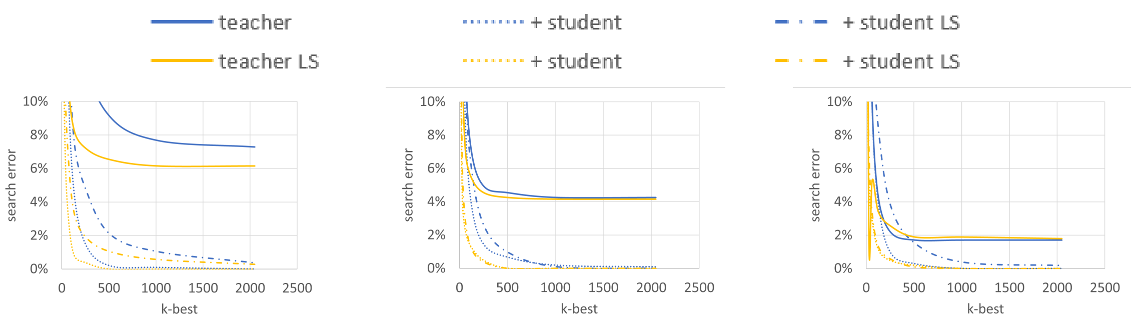
Table 8: Translation quality (BLEU) & search errors (percentages) when using LSH ($k = 1024$, $c = 2048$).

	Beam 1			Beam 4		
Model	de-en	fr-en	es-en	de-en	fr-en	es-en
Teacher with bias	28.9	31.0	40.5	29.9	31.6	41.4
+ Student no bias	29.1	31.3	40.1	29.9	31.8	40.9
+ Student no bias LS	29.6	31.6	41.4	30.4	32.5	42.0
Teacher with bias LS	29.7	31.4	41.3	30.3	32.4	42.2
+ Student no bias	28.7	31.2	40.8	29.3	31.8	41.3
+ Student no bias LS	29.6	31.5	41.0	30.2	32.1	41.9
Teacher no bias	29.1	31.3	41.0	29.8	31.8	41.5
+ Student no bias	30.1	31.4	41.8	**30.9**	32.1	42.2
+ Student no bias LS	**30.2**	32.2	41.8	30.8	32.4	42.5
Teacher no bias LS	29.2	31.2	41.1	29.9	32.3	41.8
+ Student no bias	30.0	**32.5**	41.9	30.4	**32.8**	42.3
+ Student no bias LS	29.9	31.9	**42.2**	30.5	**32.8**	**42.7**

Table 9: Translation quality of distilled models on held-out test set (BLEU) with different beam widths.

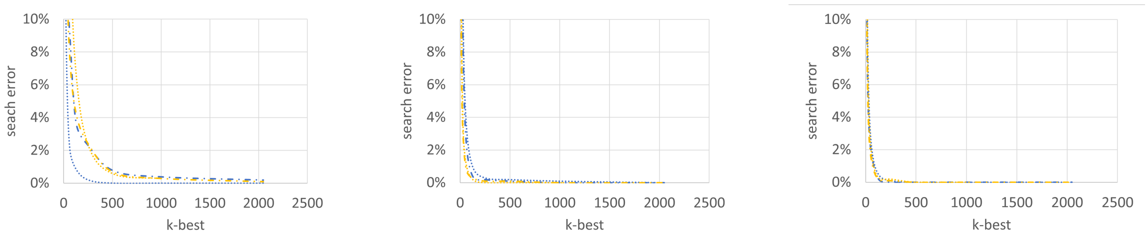
Our results thus far have been on language pairs with English as the target language. We trained and finetuned English to German models, both without output bias and label smoothing. Figure 17 shows that LSH vocabulary selection works just as well when German is the target language.

[6] github.com/marian-nmt/marian



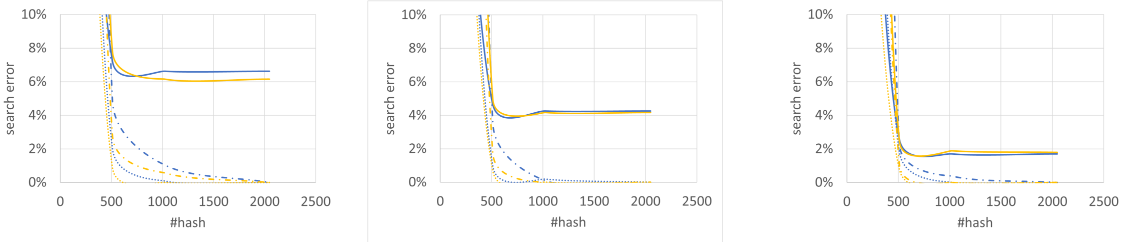
(a) de-en (b) fr-en (c) es-en

Figure 9: LSH k-best vs. search errors for models without output bias.



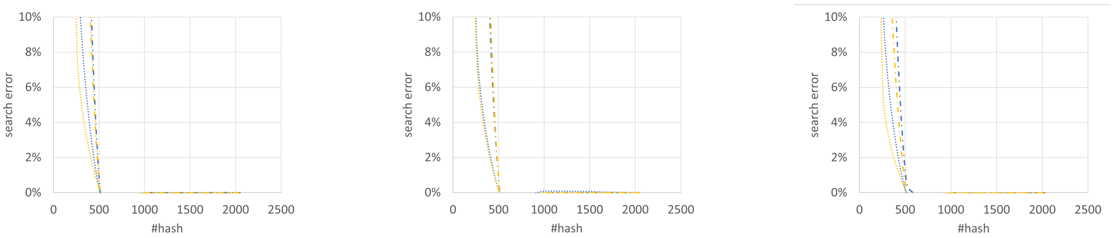
(a) de-en (b) fr-en (c) es-en

Figure 10: LSH k-best vs. search errors for models with output bias.



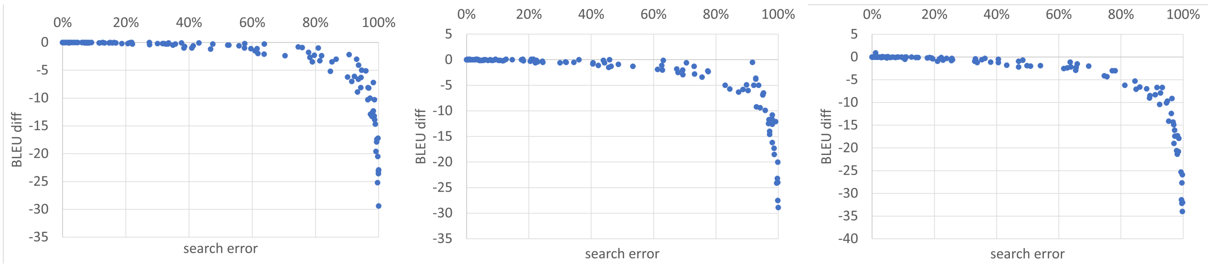
(a) de-en (b) fr-en (c) es-en

Figure 11: LSH #hashes vs. search errors for models without output bias.



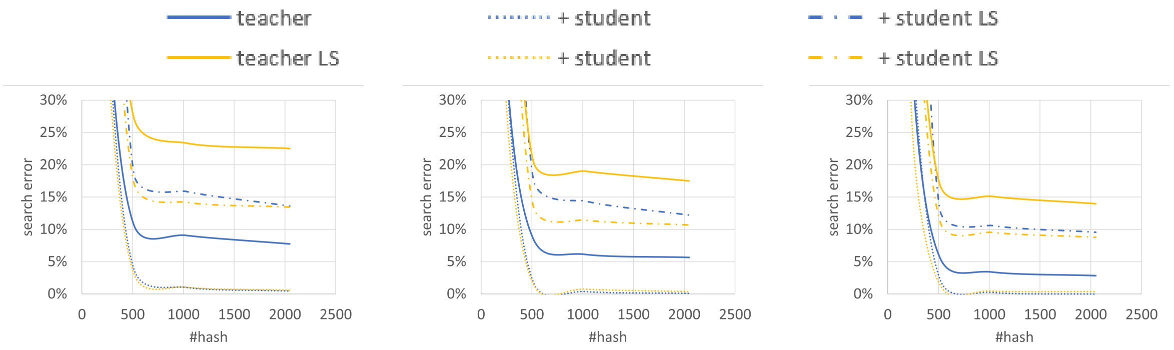
(a) de-en (b) fr-en (c) es-en

Figure 12: LSH #hashes vs. search errors for models with output bias.



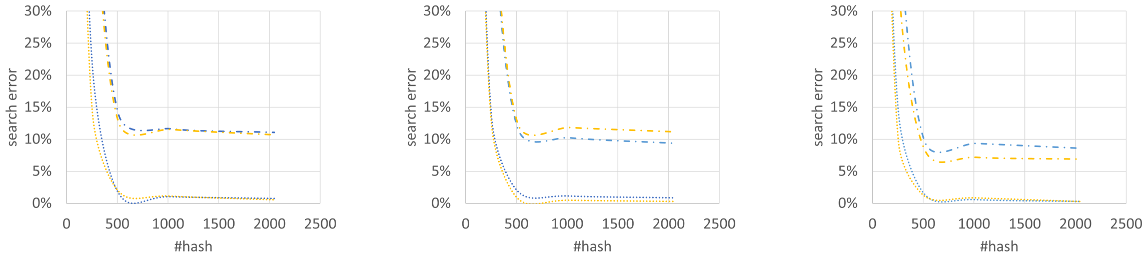
(a) de-en (b) fr-en (c) es-en

Figure 13: Search errors vs. decrease in BLEU.



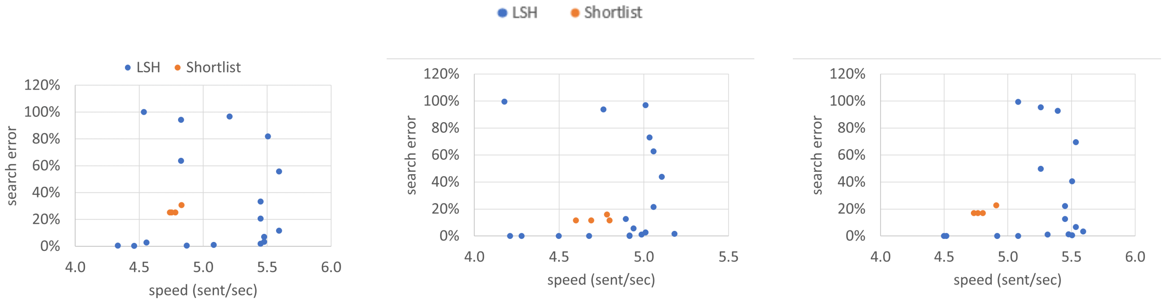
(a) de-en (b) fr-en (c) es-en

Figure 14: LSH #hashes vs. search errors for teacher models without output bias using beam width of 4.



(a) de-en (b) fr-en (c) es-en

Figure 15: LSH #hashes vs. search errors for teacher models with output bias using beam width of 4.



(b) fr-en (c) es-en

(a) de-en

Figure 16: Comparison of translation speed (sent./sec.) vs search error between LSH and lexical shortlisting.

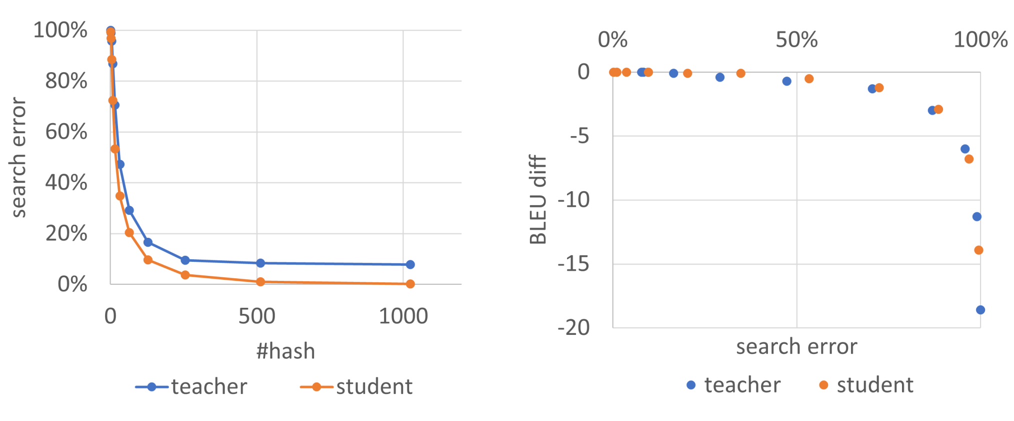


Figure 17: English-German results

Too Brittle To Touch: Comparing the Stability of Quantization and Distillation Towards Developing Lightweight Low-Resource MT Models

Harshita Diddee[1] **Sandipan Dandapat**[2] **Monojit Choudhury**[1]
Tanuja Ganu[1] **Kalika Bali**[1]
[1] Microsoft Research, India
[2] Microsoft R&D, India
{t-hdiddee,sadandap,monojitc,taganu,kalikab}@microsoft.com

Abstract

Leveraging shared learning through Massively Multilingual Models, state-of-the-art machine translation (MT) models are often able to adapt to the paucity of data for low-resource languages. However, this performance comes at the cost of significantly bloated models which are not practically deployable. Knowledge Distillation is one popular technique to develop competitive lightweight models: In this work, we first evaluate it's use to compress MT models focusing specifically on languages with extremely limited training data. Through our analysis across 8 languages, we find that the variance in the performance of the distilled models due to their dependence on priors including the amount of synthetic data used for distillation, the student architecture, training hyper-parameters and confidence of the teacher models, makes distillation a brittle compression mechanism. To mitigate this, we explore the use of post-training quantization for the compression of these models. Here, we find that while distillation provides gains across some low-resource languages, quantization provides more consistent performance trends for the entire range of languages, especially the lowest-resource languages in our target set.

1 Introduction

While NLP has made giant strides in producing more accurate models, these benefits are often not transferred representatively to end-users who would eventually use a language-technology (Ethayarajh and Jurafsky, 2020; Caselli et al., 2021). Bloated sizes, cumbersome inference times (Tao et al., 2022a) and a limited set of languages that these models serve are a few reasons for this. More specifically, their usage is hindered by access bottlenecks such as (a) **Infrastructural Obstacles**: A large percentage of end-users do not have sustained access to internet or high-compute devices to enjoy a stable access to cloud-inferencing of current NLP models (Ranathunga and de Silva, 2022; Diddee et al., 2022), (b) **Latency Requirements**: Certain NLP services (chat-bots, real-time assistance interfaces, etc.) require very low-inference time which requisite lightweight-models (c) **Privacy Constraints**: The outflow of sensitive user data which is fed for inferencing to remotely hosted NLP models also has well documented issues (Srinath et al., 2021; Huang and Chen, 2021; Huang et al., 2020; Diddee and Kansra, 2020).

Within the research that focuses on evaluating and mitigating these practical constraints, the focus on low-resource language setups has been fairly limited (Ganesh et al., 2021). For instance, while the compression of large language models has received consistent attention through analysis of pruning (Behnke and Heafield, 2020; Behnke et al., 2021), distillation (Bapna et al., 2022; Mghabbar and Ratnamogan, 2020; Kim and Rush, 2016; Junczys-Dowmunt et al., 2018) and even quantization (Bondarenko et al., 2021; Zadeh et al., 2020) - much of this work has focused on compressing language models for high-resource languages.

In this paper, we report the results of a comparative analysis of the performance of distillation and quantization. By focusing on compressing seq2seq multilingual models across a range of languages with data ranging from 7000 to 3M samples - we especially demonstrate the different priors that need to be ascertained for the successful distillation of the model. We are unaware of any previous study that demonstrates the performance of these mechanisms on such low resource languages.

The utility of this work is in commenting on the feasibility of these two compression techniques for rapid development and deployment of MT Models for low resource languages (Joshi et al., 2020). More specifically, we believe that distillation's reliance on several priors can be addressed naively through a resource-intensive exercise, where the optimal values of these priors are computed exhaustively. However, in the absence of such a budget, we expect this to be a major impediment in the development of lightweight models for such languages. Since low resource language communities may also be marginalised in other ways, exhaustive investment of data and compute might not be feasible for such communities as well as the language technologists working on these languages (Zhang et al., 2022; Diddee et al., 2022; Markl, 2022).

The main contributions of this work are:

1. We distill competitive baseline models for 8 low-resource languages (Bribri, Wixarica, Gondi, Mundari, Assamesse, Odia, Punjabi and Gujarati) and evaluate the sensitivity of the generated models to priors including (a) amount of synthetic Data being used for training (b) The architecture of the student model (c) the training hyper-parameter configuration and (d) the confidence of the teacher models.

2. We, then, quantize these models to observe if quantization provides a more consistent compression mechanism for these languages. Based on our analysis, we conclude that the suprising stability of naive Post-Training Quantization, especially in the compression of extremely-low resource languages (training data between 5000 and 25000 samples) over distillation.

We release a combination of lightweight, offline support MT models for these languages along with the scripts for generation and offline inference to further reproducible research in this domain[1].

2 Approach - Model and Size Adaptations

In this section, we describe the languages (2.1), architectures under consideration (2.1), the adaptations that we make for training and fine-tuning these models (2.2) and the adaptations we make to compress their size.

2.1 Languages

We perform our analysis on the eight languages shown in Table 1. These languages cover a wide range of availability of monolingual and parallel data, spanning from classes 0 to 3 as defined in Joshi et al. (2020). Additionally, they differ in scripts and their inclusion in pretraining corpus which result in interesting modelling adaptions that are needed to be performed for the development of their baselines. In this work, we only study the *High-Resource Language* (HRL) → *Low-Resource Language* (LRL) translation direction. The source languages for all our target languages are mentioned in Table 1.

Family of Models For this work, we leverage two model classes to carry out our analysis: **I)** seq2seq transformer (Vaswani et al., 2017), hereafter referred to as vanilla transformer: With 6 Encoder and Decoder Layers, Vocabulary size - varying between 8k to 32k and 8 attention heads. and **II)** mT5-small (Xue et al., 2021): With 8 Encoder and Decoder Layers, Vocabulary Size - 250100 and 6 attention heads.

We train the vanilla transformer from scratch, hereafter referred to as transformer, to develop a naive baseline for our experiments, and further finetune the mT5-small, hereafter referred to as mT5, with certain adaptations for all the languages, as discussed in section 2.2.

For ease of reporting, we define the highest-performing-model (denoted by HM) over our family of models as:

$$\mathrm{HM} = \operatorname*{argmax}_{\mathrm{M}} \mathrm{A(M)}$$

where M is a model class with performance A(M) after training (where A is a metric like BLEU (Papineni et al., 2002) or chrF (Popović, 2016) used to monitor the task-specific performance of the model).

2.2 Model Adaptations: Language Specific Approaches

Here we describe the strategies required to adapt these models to different low-resource languages: During fine-tuning, we adapt the pretrained mT5 tokenizer to unseen scripts (encountered for Odia) by transliterating it to the closest, highest-resource language included in the pretraining corpus of the pretrained model (Khemchandani et al., 2021; Ramesh et al., 2021, 2022). For our extremely low-resource languages, we used Lexicon-Adaption (Wang et al., 2022) for the augmentation of target-side monolingual data for languages wherever a bilingual lexicon could be leveraged - Detailed performance with Hindi-Gondi is provided in the Appendix section A.2. However since such methods were not extensible to all the languages in our target language set, we report final experimental results on the models

[1] Codebase and Open-Sourced Models

Language	Class	Source Language	Data Constraints		Model Constraints	
			Monolingual Data	Parallel Data	Shared Script	Included in Pretraining
Bribri	0	Spanish	✗	✓	✗	✗
Wixarica	0	Spanish	✗	✓	✗	✗
Mundari	0	Hindi	✗	✓	✓	✗
Gondi	0	Hindi	✗	✓	✓	✗
Assammese	1	English	✓	✓	✓	✓
Odia	1	English	✓	✓	✗	✗
Punjabi	2	English	✓	✓	✗	✓
Gujarati	1	English	✓	✓	✗	✓

Table 1: Languages Under Consideration: Note that the except the language's inclusion in the pretraining corpus of our chosen pretrained language models, all factors are independent of our experimental setup. Source language column enlists the source language of the translation pairs

which did not leverage any additional data other than the data mentioned in A.1. Since we analyze the HRL to LRL direction and 4 out of 8 (Bribri, Wixarica, Gondi and Mundari) of our target languages have little to negligible monolingual data - we were also unable to leverage Back-Translation to augment our language-specific parallel corpus (Edunov et al., 2018).

2.3 Size Adaptation: Knowledge Distillation

Knowledge distillation involves training a smaller student network to mimic the token level probabilities of a larger, more accurate teacher model. We distill our models using Hard Distillation (Kim and Rush, 2016): we utilize a set of monolingual sentences in the HRL - and forward translate using the HM to generate synthetic labels that a lighter student model is then trained on.

2.3.1 Estimation of Optimal Values for Priors

We define a prior as any attribute of the compression mechanism that needs to be initialized meaningfully and/or optimized for optimal performance - akin to hyperparameters. We use this term specifically so as to put all the dependent variables - such as training data, prediction confidence of the uncompressed models, etc in a single bucket: rather than using a term like hyperparameters that already holds traditional significance in literature. The experimental sweeps for these priors are briefly explained in this section. Note that we focus largely on distillation while estimating for these priors, because quantization provides competitive models even with the default choices established by literature whereas with distillation - the estimation of these priors is critical to achieve a competitive compressed model variant in most cases.

Prior 1: Optimal Student Architecture Following prior work like Bapna et al. (2022), we experimented with 3 candidate architectures, two of which used deep encoders and shallower decoders. We sweeped across 3 candidate architectures - all variants of a seq2seq transformers with (a) 8 Encoder + 6 Decoder Layers (b) 6 Encoder + 4 Decoder Layers and (c) 6 Encoder + 3 Decoder Layers. We chose the architecture that gave the best BLEU performance after 30 epochs. Sweeps for the architecture were done across each of the following languages - Gondi, Assamesse and Odia as they covered a wide range of training data.

Prior 2: Optimal Training Hyperparameters We sweeped across a set of hyper-parameter sets for Bribri, Gondi, Assamesse and Gujarati to identify the optimal set for the distilled student models. Our goal here was to specifically study the transferability of a hyperparameter set which performed competitively for one or more languages, to all the languages in our target set.

Prior 3: Amount of Training Data for the Student We sweeped across 3 candidate sizes of our synthetic dataset: 100K, 250K and 500K pseudo-labels. Since this decision could also be greatly dependent on the quality of the labels generated per language - we ran this sweep for Bribri, Gondi, Odia and Gujarati, as the quality of the labels generated by the teachers for these languages would be expected to demonstrate significant variation.

Prior 4: Optimal Teacher Architecture To do a preliminary quantification of the effect of the choice of a teacher architecture and the quantity of data that a teacher is trained for on the compressibility of the model - we decided to evaluate the confidence of our teacher models on the predictions they generated. For this, we sampled 100 instances from each of our testsets and monitored the logit distribution of our teacher models. Specifically, we calculated the average of the softmax entropy of the token-level softmax distributions for a sequence. Taking inspiration from the unsupervised estimation of quality of machine translation outputs (Fomicheva et al., 2020) through similar methods, we hypothesised that the lower the entropy of our model, the more confident it would be in its predictions for a given sample. The intuition here was that if a model is confident about its prediction, its logit distribution would be highly-skewed, and not resemble a uniform distribution (which would indicate its indecisiveness in being able to predict the right token - and therefore, the right sequence). Eventually, this could be used to gauge the quality of the pseudo labels that are student were being trained on.

2.4 Size Adaptation: Quantization

Quantization is a common way to reduce the computational time and memory consumption of neural networks (Wu et al., 2020). Here, a lower-bit representation of weights and activation functions is used to achieve a lower memory footprint. In this work, we perform post-training quantization, where after training the base model with full precision of floating point 32 bits (fp-32), we convert the weights and activations of the model to 8 bit integers (int-8). Note that during inference, we still preserve the precision of the input and output encoder-decoder distributions as fp-32. In theory, this brings down the memory consumption of the model by nearly 4x times, though we see an effective reduction of about 3x in practice. More details on the memory-reductions achieved are specified in the Appendix A.4

3 Experimental Setup

3.1 Data

(a) Bribri and Wixarica: We use the training data 7K and 8K sentences, respectively from Feldman and Coto-Solano (2020) and evaluate on test data from Mager et al. (2021). **(b) Gondi**: We use 26k sentences from the data opensourced by CGNET Swara (CGNET, 2019) and split it into training and test sets.[2] **(c) Mundari:** We use a dataset of 10K sentences provided by Indian Institute of Technology, Kharagpur[3], and split it into training and test sets.[1] **(d) Assamesse, Odia, Punjabi and Gujarati**: We use the training data from Ramesh et al. (2022) (with 0.14M, 1M, 2.4M and 3M sentences, respectively) and evaluate on test data from FLORES200 Goyal et al. (2022) for Assamese and WAT2021 Nakazawa et al. (2021) for the remaining languages. Additional details about datasets (sizes and splits) are mentioned in the Appendix A.1.

3.2 Training Setup

Hyperparameters: We use the transformer and mT5 as our model classes as described previously in Section 2. The hyperparameters for our transformer model was optimized for fine-tuning of Odia, trained on 1M sentence pairs. For fine-tuning, we use the Adafactor optimizer (Shazeer and Stern, 2018), with a linearly decaying learning rate of 1e-3. Since training with smaller batches is known to be more effective for extremely low-resource language training (Atrio and Popescu-Belis, 2022), we tuned the training batch size for every language - varying from 32 to 256 (with gradient accumulation as 2) though we did not see very significant variation in the performance on the basis of this tuning. For our stopping criteria: we fine-tuned all models for 60 epochs (which concluded with considerably overfit models) and then selected models by we picking the checkpoint which had the best validation performance on BLEU (with only the 13a tokenizer which mimics the mteval-v13a script from Moses) (Post, 2018).

We use the sentencepiece tokenizer to build tokenizers for training the baselines for each of the languages (Kudo and Richardson, 2018). We use the per-token cross-entropy loss for fine-tuning all our models. Following Xu et al. (2021), we opt for a relatively smaller vocabulary size with the intent of learning more meaningful subword representations for our extremely low-resource languages. Specifically, we use a vocabulary size of 8K for Gondi, Mundari, Bribri and Wixarica, compared to 32K used for Assamesse, Odia Punjabi and Gujarati.

Experimental Setup for Distillation For Mundari and Gondi we utilize 500K Hindi sentences sampled from the Samanantar corpus (Ramesh et al., 2022); We use the corresponding English corpus to sample English sentences for generating the pseudo labels for Assamesse, Odia,

[2] To avoid any test-set leaks, we deduplicate the data by removing tuples (S^i, T^i) where S^i is the i^{th} sentence in the source language and T^i is $i^{th}$the sentence in the target language, between the train and the test set.

[3] Data to be released soon;

Punjabi and Gujarati. For Bribri and Wixarica - We use Spanish data made available by the Tatoeba Challenge (Tiedemann, 2020). We use the per-token cross-entropy loss for training our distilled models.

Evaluation Metrics: We use BLEU (sacrebleu with spm pre-tokenization (version 2.2.0)) (Post, 2018) for all our evaluations (Goyal et al., 2020). In addition to this, we also report chrF2 (Popović, 2016) for all our experiments for a more comprehensive comparison between the models.

4 Results

In section 4.1, we present the performances of our base models in Table 2. In the following section 4.2, we report the performances of the distilled HM in Table 3. Using these empirical results we focus on answering the following questions (a) To what degree can scaling the student training data improve the performance of the student model? (4.3) (b) How sensitive is distillation to the choice of the architecture of the student model? (4.4) (c) How can we choose an optimal teacher that is most suitable for compression? (4.5) (d) To what degree does the hyperparameter set suitable for distilling a model for one language transfer to another language? (4.6)

While answering these questions, we also analyze in parallel the performance of the quantized variants of these models implicitly indicating the reduced sensitivity of these variants from most of the previously discussed priors in spite of their competitive performances.

Language	Data	Vanilla transformer		mT5	
		spBLEU	chrF2	spBLEU	chrF2
Bribri	7K	1.7	11.6	**6.4**	19.3
Wixarica	8K	2.2	14.1	**6.2**	28.0
Mundari	10k	0.1	5.6	**15.9**	33.7
Gondi	26K	1.2	7.9	**14.3**	32.5
Assamesse	140K	0.8	12.4	**10.7**	30.4
Odia	1M	23.7	43.6	**27.4**	47.6
Punjabi	2.4M	**38.4**	50.6	34.8	44.1
Gujarati	3.05M	**35.9**	53.4	35.7	49.8

Table 2: Performance of our base models (transformer and mT5) without quantization or distillation. Best performing models out of the two architectures are marked in bold.

4.1 Analyzing the Baseline Models

As expected, the transformer models for target languages start competing (and outperforming) once an adequate amount of data is available for training the vanilla transformers. In addition to the obvious gain for being only optimized for target languages, the performance gains of these baselines can also be attributed to the language-specific tokenizer that they utilize, in contrast to the pretrained mT5 tokenizer that might be sub-optimal for language-specific generation. For our low-resource languages though, the advantage of transfer learning is clearly evident: all languages achieve a minimum and maximum performance improvement of 4 and 16 BLEU points. Gondi and Mundari, despite having relatively low-amount of data, perform well - though we expect an overestimation of their performance due to the homogenity between the train and the test set. Additionally though, both languages share scripts with a dominant language script i.e., Devanagari and hence, can be expected to gain because of that.

4.2 Analyzing the Compressed Models

In Table 3, we briefly present the performances of our distilled and quantized models. As evident, especially for the lowest-resource models, both distillation and quantization give competitive performance in addition to providing a significant size reduction. Note that Table 3 does not report the performance of the quantization of the vanilla transformer models for Odia, Gujarati and Punjabi even though they had competed or outperformed the mT5 variants. This is because they suffered a significant drop in performance - Odia dropped in performance to 8.4 BLEU/30.5 chrF2 in contrast to its HM scores of 23.7 BLEU/ 43.6 chrF2 respectively. Gujarati and Punjabi also dropped to 16 BLEU/31.2 chrF2 and 19.1/36.0 , respectively. To explain this we note what distinguishes these two architectures: (a) mT5 is deeper than transformer having 2 extra layers on the encoder's side than the vanilla transformer and (b) leverages multilingual pretraining. These attributes become useful in interpreting mT5 robustness to compression. In agreement with prior work like Li et al. (2020), deeper models can be expected to be more immune to compression. In fact, these models can be expected to be regularized by a certain degree through quantization, and we posit that we might be adopting a sub-optimal fine-tuning hyperparameter set for the initial fine-tuning of these models, consequently generating potentially overfit models and this gets mitigated to some extent upon quantization. Taking into consideration the lack of prior work on fine-tuning large LMs on such extremely low-resource languages and the infeasibility of running intricate hyperparameter sweeps per language with such large models, this can also be expected to degrade the quality of the labels generated for training the distilled models - ultimately affecting the performance that the distilled models achieve.

Language	HM	Distilled HM		Quantized HM	
	spBLEU	spBLEU	chrF2	spBLEU	chrF2
Bribri	6.4	6.8	13.2	**7.4**	19.4
Wixarica	6.2	4.1	17.3	**7.2**	26.8
Mundari	15.9	**18.2**	32.7	15.7	29.3
Gondi	14.3	**14.2**	32.8	13.8	31.1
Assamesse	10.7	**9.6**	27.4	6.2	25.7
Odia	27.4	20.2	40.7	**21.0**	41.3
Punjabi	38.4	**32.8**	46.6	27.0	48.0
Gujarati	35.9	**29.8**	48.6	28.4	51.4

Table 3: Performance of the HM for all languages after applying Distillation and Quantization. Best performing models out of both of the size adaptations are marked in bold.

In the following sections we focus on presenting our analysis of distillation's sensitivity to certain priors. In each section, we also discuss an analysis of the same priors' effect on quantization. Note that since the mT5 outperformed the vanilla transformer variants for all languages up till Odia - we distilled and quantized them for these languages. Also note that the HM for these languages is hence, mT5. Additionally, for Odia, Gujarati and Punjabi, we quantized both the mT5 and the vanilla transformer variants of the models.

4.3 Sensitivity to Priors: Data

The quality, quantity and the domain of data that the teacher or uncompressed variant of the model is trained on, appears to impact both the mechanisms of compression: For distillation the gold training data as well as the monolingual data utilized for generating student labels is of relevance, and for quantization only the gold data that the teacher is fine-tuned for, is of relevance.

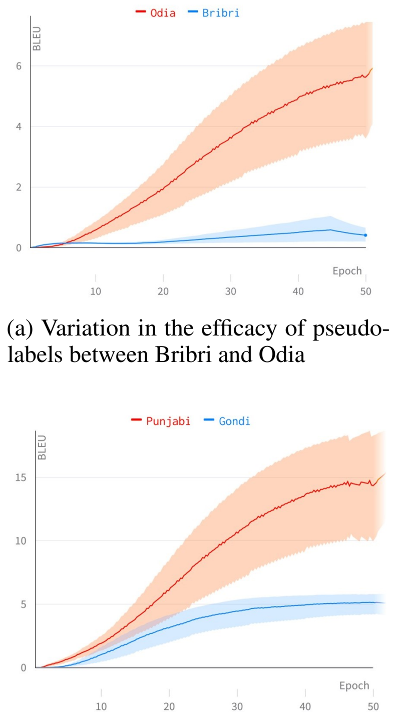
Quantity of Training Data Interestingly, quantization displayed consistent performance variations across the entire range of our low-resource language sets (all languages up till Odia), giving marginally close scores to the HM so at least within the data sparse languages we did not see any direct variation in the performance according to the amount of training data used. Both mechanisms show nearly equal degradation in performance for the HRL.

Quality of Training Data The quality of the data that the teacher is trained on affects the model's immunity to compression. This is best demonstrated by the post-compression performances of Gondi and Mundari in Table 3: In Gondi - the train set has nearly 26K sentences, which by the virtue of being collected via crowd-sourcing may be expected to be noisy. Mundari's training data, though also crowd-sourced, claims to have been validated manually after its collection by the providers to generate the final corpus of about 10K sentences. The observed difference where Gondi suffers a slight performance degradation post-compression and Mundari experiences a significant performance gain, may be attributed to the difference in the quality of their training data. Note that both languages are being translated from the same source language, share the same script and are being tested on a correlated test set - so the quality and quantity of training data are expected to be major contributors to the variations in their performance.[4]

Quantity of Pseudo-Labels used for Student's Training Results of our analysis of scaling student data between 100K to 500K are presented in Figure 1. More data seemed to help for the entire spectrum of languages - though it is evident that the gain in the performance diminished in proportion to the amount of added data as we approached the lowest-resource languages in our set. The gain in performance upon the addition to 250K samples to a HRL like Odia or Punjabi is significantly more pronounced than the gain in performance for Bribri or Gondi - where there is a very marginal improvement in the performance upon the addition of 250K samples. This could be indicative of the diminishing efficacy of the increasingly noisy data that was generated by the lowest-resource teachers. We explore this notion in more depth in Section 4.5.

Domain of Data While we do not perform any targeted experiments to evaluate the domain dependence of the two compression mechanisms - we posit that the distilled models' significantly better performance than its quantized variant in As-

[4] The two languages do belong to two different language families - Gondi belonging to the Dravidian language family which has a higher representation in the pretraining corpus for mT5, and Mundari being Austro-Asiatic



(a) Variation in the efficacy of pseudo-labels between Bribri and Odia

(b) Variation in the efficacy of pseudo-labels between Punjabi and Gondi

Figure 1: Min/Max range curves of the performance of the models trained on scaled data: The shaded range is considerably lower for the lowest-resource languages indicating reduced efficacy of scaling student data.

samesse could be attributed to the distilled model's exposure to the diverse-domain data during the student's training. Note that the testset used in Assamesse, FLORES 200 (Goyal et al., 2022), is claimed to be of a very diverse-domain origin. Given this, the process of training a student on monolingual data of a potentially more diverse origin to that of the native training set - would explain the gain that the language demonstrates on a domain-agnostic testset. Prior work like Mghabbar and Ratnamogan (2020) already shows distillation's efficacy in enabling students to adapt to out-of-domain data that the teacher may not have ever been exposed to. Quantization on the other hand, has no opportunity for exposure to any out-of-domain data - so its adaptation and performance across a domain-agnostic testset can be expected to only degrade.

4.4 Sensitivity to Priors: Student Architecture

We find that distilled student models could be adversely sub-optimal for a given language, despite being sub-optimal or even an optimal choice for a large subset of languages. To demonstrate this

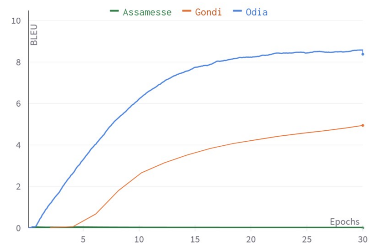


Figure 2: Variation in BLEU due to difference in the choice of a student architecture: An optimal architecture choice for Odia and Gondi gives adversely sub-optimal performance for Assamesse

in Figure 2, we show the performance of two distilled models on an identical hyperparameter set and student architecture. While the chosen student architecture gives competitive performances for Gondi and Odia, Assamesse performs significantly worse for this candidate architecture. We did attempt retraining the model with a different seed to negate the possibility of a randomly poor initialization though this did not improve the convergence. While we did not notice such a drastic performance variation across any other candidate set, this instance did indicate brittleness to the student-architecture for a given language. After these sweeps, we fixed a transformer-based encoder with 6 layers and a transformer-based decoder with 4 layers as the distilled model for our further experiments.

4.5 Sensitivity to Priors: Confidence of the Teacher Model

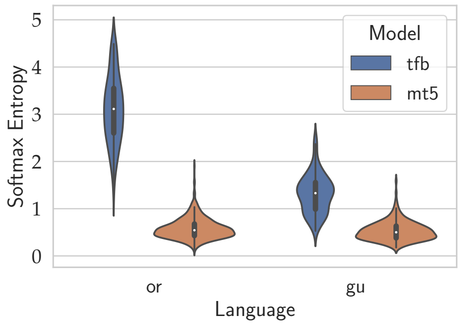


Figure 3: Entropy distributions of mT5 and transformer: lower-entropy indicates high-confidence and consequently suggest higher-quality of translations.

Estimating the confidence of our teacher models displayed manifold benefits: Within Distillation, it helped us get an indirect estimate of the quality of the training data that the student model was trained on. Within Quantization, it was useful in analyzing why the mT5-variants were more robust to quantization. Note that since the testsets for all the languages are of varying difficulty - doing a language-wise comparison on the basis of such metrics was non-trivial since the confidence predictions could also vary in accordance with the complexity of the testsets being evaluated upon. Hence, we majorly focused on analyzing languages which were either evaluated on the same test set (Gujarati, Punjabi, Odia with WAT21 testset (Nakazawa et al., 2021)) or the different architectures for each of our languages which could be evaluated for the same testset.

Figure 3 demonstrates the difference in the entropy of the softmax distributions of the mT5 and transformer teacher variants. Note that this is for Gujarati and Odia, our highest resource language, for which both architectures perform quite competitively and the vanilla transformer even outperforms the mT5.

As is evident, the mT5 variant has much lower entropy scores, with lower dispersion indicating high-confidence in the predictions it produces for each of the samples. Note that the inference pipeline for both architectures is identical - Greedy Search with no sampling so we don't expect any difference in the decoding mechanism to affect the quality or distribution of representations that we are monitoring. This is a very interesting observation, as both models appear to perform comparably according to our automatic metric evaluations - yet differ quite significantly in the stability with which they generate these predictions.

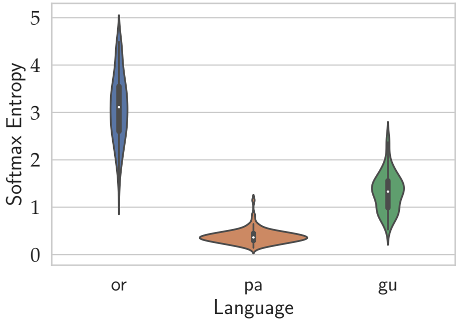


Figure 4: Entropy distributions for transformer across different languages: Models become increasingly more confident about their predictions with an increase in training data

Next, we attempt to establish if training with more data makes a model more confident in its prediction. Figure 4 demonstrates the entropy scores for Odia, Punjabi and Gujarati. Each of these have data increasing in the order of 1M, 2.4M and 3M respectively. Here we observe that indeed, models trained with more data achieved consistently lower entropy scores.

4.6 Sensitivity to Training Hyperparameters

In this section we present results of evaluating if an adequate hyperparameter set for a given language may be suitable for generating an optimal variant for another distilled language. Here too, we demonstrate using a subset of our hyperparameter sweep that there can be a marked degradation in the suitability of an averagely optimal hyperparameter set (that might be close to optimal to multiple languages with similar attributes) to an unseen language;

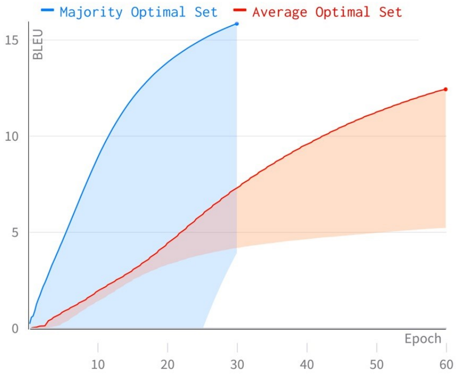


Figure 5: Min/Max range of performances of Gujarati, Bribri and Assamesse across a hyperparameter set that is optimal for these languages but adversely sub-optimal set for Gondi

In Figure 5, when tuned for the hyperparameter set that is optimal for a majority of languages in our set, Gondi does not even converge as a result of which the lower-bound of a teacher's performance for that hyperparameter set is 0. Note that this hyperparameter set transferability does not seem to show any specific data oriented trends as well. For instance, the same hyperparameter set that was optimal for Gujarati, our highest resource language with 3M data points, is only slightly sub-optimal for Bribri, our lowest resource language with 7000 data points, and Assamese, our mid-resourced language with 135K sentences. Also note that we were able to get acceptable performance for Gondi with almost an identical hyperparameter setup with a larger batch size (quadrupled to the one in this setup) indicating that a per-language sweep would be an ideal and acceptable solution even though this would imply that distilling models would mandate a significant hyperparameter tuning for achieving optimal performance. A detailed list of what hyperparameters we sweeped through can be found in the Appendix Table 5.

5 Takeaways

We encapsulate the learning from our analysis as the following takeaways:

1. ***Data Dependence of the Method of Compression:*** Training teacher models with lesser quantity, higher quality data is expected to improve a model's robustness against both quantization and distillation. The post-quantization performance suffers equally for models trained with varying degrees of data. This is not the case with distillation, where increasing the amount of training data for student distilled models starts providing diminishing returns as the amount of training data for the teacher reduces.

2. ***Cost of Compression:*** Distillation is quite sensitive to its training hyperparameters and the student's architecture. This choice doesn't necessarily follow any data-oriented trends as well i.e., languages having similar amount of data may perform very differently on similar hyperparameter and student architecture sets. Hence, Distillation mandates a significant hyperparameter tuning cost that Quantization does not incur.

3. ***Stability of Compression*:** Hard Distillation and Post-Training Quantization are both promising methods of quickly compressing massively multilingual models for machine translation for extremely low-resource languages. Post-Training Quantization should be preferred when the uncompressed variants is pretrained and/or deep, expected degree of compression is upto 4x the original model's size and the cost of compression is to be minimum. Distillation, on the other hand, should be preferred when domain-expansion, language-specific tokenization and more than 4x degree of compression needs to be achieved at the cost of a tuning for optimal architecture and training setup selection.

6 Related Work

Owing to the known benefits of compressing language models due to their lower-memory footprint, improved inference speed and even improved performance in some cases, compression techniques have been explored widely in NLP.

Quantization While the work on quantizing encoder-models is replete (Zafrir et al., 2019; Bondarenko et al., 2021; Kim et al., 2021; Zadeh et al., 2020) the focus on quantizing decoder-only models (Tao et al., 2022b), and specifically seq2seq models has been relatively much lower. Recent work like, EdgeFormer, (Ge and Wei, 2022), LLM.int8() (Dettmers et al., 2022) have recently demonstrated the generation of seq2seq quantized models which provide a high-compression ratios and competitive performances though this work has also been done with much higher resource languages.

Distillation Work within distillation is replete, even for the multilingual-type of models that we focus on. Work like Kaliamoorthi et al. (2021); Jiao et al. (2021); Yang et al. (2022) represent the major body of work in multi-lingual distillation - that is also centered across the encoder-only space. Relatively lesser work has been done in the space of mutli-lingual distillation (Soltan et al., 2021; Mukherjee et al., 2021) of seq2seq models and even though work like Zhang et al. (2020); He et al. (2019) extends this analysis to relatively low-resource languages, they rely on the use of monolingual data for the target language, a luxury that we cannot afford for half of the languages in our language set.

Note that since both processes are orthogonal, their conjunctive use has also been explored - Tao et al. (2022a) for instance, get competitive results by applying token level contrastive distillation and module-wise dynamic scaling while quantizing generative models. Note that we made the conscious decision of excluding pruning from our analysis because while it is known to demonstrate very effective parameter reduction, it is generally not as aggressive in it's memory footprint reduction as much as quantization and distillation (Behnke and Heafield, 2020; Mohammadshahi et al., 2022). As we'll discuss further in section 7, size-reduction was an implicit focus of this work that is one of the most fundamental bottlenecks of community deployment A.4.

7 Discussion

While this work explicitly focuses on only the performance comparison between distillation and post-training quantization, it's efficacy can also be viewed in demonstrating the development of lightweight, machine translation models for extremely low-resource languages. This is a very critical outcome as Performance-oriented Machine translation (MT) models for low-resource languages are often not suited for the immediate consumption of the community. The access bottleneck introduced by these bloated models, can especially affect those communities which haven't traditionally enjoyed access to a digital ecosystem, often widening the gap between those who can and cannot access these tools. Towards this direction, the exploration of compression strategies for these models - especially when tied to end-user centric NLP services such as translation is imperative. In this work, the size of all models being evaluated after compression was less than 400MB - the quantized models are at least 3x lighter the size of the native HM and the distilled models give even more impressive gains of upto 8x smaller than their uncompressed counterparts. This size reduction, coupled with the increased speed of inference associated with this reduction in most cases can enable a suite of accessible translation models for these languages[5]. This establishes a very promising potential in achieving deployment-constraint aware models: For instance, in areas where users do not enjoy a sustained access to the internet - these lightweight models may be adapted to operate on edge in an offline fashion.

8 Conclusion and Future Work

In this work we established that hard-distillation is sensitive to several priors which makes it a brittle mechanism of compression, especially for languages with extremely low-resources. In relative comparison, post-training quantizaton provides a competitive, stable and cost-effective compression mechanism that works effectively for extremely low-resource languages as well. Moving forward, we wish to explore the effect of using additional data (augmented or natively available) on the compressed variants of these models and extend distillation's analysis to utilizing logit distributions of the teacher (soft-distillation). Having observed the poor confidence measures of the transformer - and it's relatively random distributions we expect to get more interpretable evidence towards the suitability of these models for soft distillation through such an analysis.

[5] A more detailed description of the sizes of these models and the associated inference patterns is provided in the Appendix A.4

Acknowledgements

We sincerely thank the reviewers for their detailed feedback on the work which greatly helped us improve the quality of this work. Additionally, we thank Indian Institute of Technology, Kharagpur for giving us access to the Mundari data. Finally, we also thank Anurag Shukla for helping formalize the quantization pipeline and Kabir Ahuja, Sumanth Doddapaneni and Naman Jain for all the helpful discussions.

References

Àlex R Atrio and Andrei Popescu-Belis. 2022. Small batch sizes improve training of low-resource neural mt. arXiv preprint arXiv:2203.10579.

Ankur Bapna, Isaac Caswell, Julia Kreutzer, Orhan Firat, Daan van Esch, Aditya Siddhant, Mengmeng Niu, Pallavi Baljekar, Xavier Garcia, Wolfgang Macherey, et al. 2022. Building machine translation systems for the next thousand languages. arXiv preprint arXiv:2205.03983.

Maximiliana Behnke, Nikolay Bogoychev, Alham Fikri Aji, Kenneth Heafield, Graeme Nail, Qianqian Zhu, Svetlana Tchistiakova, Jelmer van der Linde, Pinzhen Chen, Sidharth Kashyap, and Roman Grundkiewicz. 2021. Efficient machine translation with model pruning and quantization. In Proceedings of the Sixth Conference on Machine Translation, pages 775–780, Online. Association for Computational Linguistics.

Maximiliana Behnke and Kenneth Heafield. 2020. Losing heads in the lottery: Pruning transformer attention in neural machine translation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 2664–2674, Online. Association for Computational Linguistics.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. 2021. Understanding and overcoming the challenges of efficient transformer quantization. arXiv preprint arXiv:2109.12948.

Tommaso Caselli, Roberto Cibin, Costanza Conforti, Enrique Encinas, and Maurizio Teli. 2021. Guiding principles for participatory design-inspired natural language processing. In Proceedings of the 1st Workshop on NLP for Positive Impact, pages 27–35, Online. Association for Computational Linguistics.

Swara CGNET. 2019. Hindi-gondi parallel corpus. https://arxiv.org/abs/2004.10270.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm. int8 (): 8-bit matrix multiplication for transformers at scale. arXiv preprint arXiv:2208.07339.

Harshita Diddee, Kalika Bali, Monojit Choudhury, and Namrata Mukhija. 2022. The six conundrums of building and deploying language technologies for social good. In ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies (COMPASS), COMPASS '22, page 12–19, New York, NY, USA. Association for Computing Machinery.

Harshita Diddee and Bhrigu Kansra. 2020. Crosspriv: User privacy preservation model for cross-silo federated software. In 2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE), pages 1370–1372.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. arXiv preprint arXiv:1808.09381.

Kawin Ethayarajh and Dan Jurafsky. 2020. Utility is in the eye of the user: A critique of NLP leaderboards. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4846–4853, Online. Association for Computational Linguistics.

Isaac Feldman and Rolando Coto-Solano. 2020. Neural machine translation models with back-translation for the extremely low-resource indigenous language Bribri. In Proceedings of the 28th International Conference on Computational Linguistics, pages 3965–3976, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. Transactions of the Association for Computational Linguistics, 8:539–555.

Prakhar Ganesh, Yao Chen, Xin Lou, Mohammad Ali Khan, Yin Yang, Hassan Sajjad, Preslav Nakov, Deming Chen, and Marianne Winslett. 2021. Compressing Large-Scale Transformer-Based Models: A Case Study on BERT. Transactions of the Association for Computational Linguistics, 9:1061–1080.

Tao Ge and Furu Wei. 2022. Edgeformer: A parameter-efficient transformer for on-device seq2seq generation. arXiv preprint arXiv:2202.07959.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. Transactions of the Association for Computational Linguistics, 10:522–538.

Vikrant Goyal, Sourav Kumar, and Dipti Misra Sharma. 2020. Efficient neural machine translation for low-resource languages via exploiting related languages. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 162–168, Online. Association for Computational Linguistics.

Tianyu He, Jiale Chen, Xu Tan, and Tao Qin. 2019. Language graph distillation for low-resource machine translation. arXiv preprint arXiv:1908.06258.

Tao Huang and Hong Chen. 2021. Improving privacy guarantee and efficiency of Latent Dirichlet Allocation model training under differential privacy. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 143–152, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yangsibo Huang, Zhao Song, Danqi Chen, Kai Li, and Sanjeev Arora. 2020. TextHide: Tackling data privacy in language understanding tasks. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1368–1382, Online. Association for Computational Linguistics.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2021. Lightmbert: A simple yet effective method for multilingual bert distillation. arXiv preprint arXiv:2103.06418.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6282–6293, Online. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Kenneth Heafield, Hieu Hoang, Roman Grundkiewicz, and Anthony Aue. 2018. Marian: Cost-effective high-quality neural machine translation in c++. arXiv preprint arXiv:1805.12096.

Prabhu Kaliamoorthi, Aditya Siddhant, Edward Li, and Melvin Johnson. 2021. Distilling large language models into tiny and effective students using pqrnn. arXiv preprint arXiv:2101.08890.

Yash Khemchandani, Sarvesh Mehtani, Vaidehi Patil, Abhijeet Awasthi, Partha Talukdar, and Sunita Sarawagi. 2021. Exploiting language relatedness for low web-resource language model adaptation: An indic languages study. arXiv preprint arXiv:2106.03958.

Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2021. I-bert: Integer-only bert quantization. In International conference on machine learning, pages 5506–5518. PMLR.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. arXiv preprint arXiv:1808.06226.

Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joey Gonzalez. 2020. Train big, then compress: Rethinking model size for efficient training and inference of transformers. In International Conference on Machine Learning, pages 5958–5968. PMLR.

Manuel Mager, Arturo Oncevay, Abteen Ebrahimi, John Ortega, Annette Rios, Angela Fan, Ximena Gutierrez-Vasques, Luis Chiruzzo, Gustavo Giménez-Lugo, Ricardo Ramos, Ivan Vladimir Meza Ruiz, Rolando Coto-Solano, Alexis Palmer, Elisabeth Mager-Hois, Vishrav Chaudhary, Graham Neubig, Ngoc Thang Vu, and Katharina Kann. 2021. Findings of the AmericasNLP 2021 shared task on open machine translation for indigenous languages of the Americas. In Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas, pages 202–217, Online. Association for Computational Linguistics.

Nina Markl. 2022. Mind the data gap(s): Investigating power in speech and language datasets. In Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion, pages 1–12, Dublin, Ireland. Association for Computational Linguistics.

Idriss Mghabbar and Pirashanth Ratnamogan. 2020. Building a multi-domain neural machine translation model using knowledge distillation. arXiv preprint arXiv:2004.07324.

Alireza Mohammadshahi, Vassilina Nikoulina, Alexandre Berard, Caroline Brun, James Henderson, and Laurent Besacier. 2022. What do compressed multilingual machine translation models forget? arXiv preprint arXiv:2205.10828.

Subhabrata Mukherjee, Ahmed Hassan Awadallah, and Jianfeng Gao. 2021. Xtremedistiltransformers: Task transfer for task-agnostic distillation. arXiv preprint arXiv:2106.04563.

Toshiaki Nakazawa, Hideki Nakayama, Isao Goto, Hideya Mino, Chenchen Ding, Raj Dabre, Anoop Kunchukuttan, Shohei Higashiyama, Hiroshi Manabe, Win Pa Pa, Shantipriya Parida, Ondřej Bojar, Chenhui Chu, Akiko Eriguchi, Kaori Abe, Yusuke Oda, Katsuhito Sudoh, Sadao Kurohashi, and Pushpak Bhattacharyya, editors. 2021. Proceedings of the 8th Workshop on Asian Translation (WAT2021). Association for Computational Linguistics, Online.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pages 311–318.

Maja Popović. 2016. chrF deconstructed: beta parameters and n-gram weights. In Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers, pages 499–504, Berlin, Germany. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting bleu scores. arXiv preprint arXiv:1804.08771.

Akshai Ramesh, Venkatesh Balavadhani Parthasarathy, Rejwanul Haque, and Andy Way. 2021. Comparing statistical and neural machine translation performance on hindi-to-tamil and english-to-tamil. Digital, 1(2):86–102.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Divyanshu Kakwani, Navneet Kumar, et al. 2022. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. Transactions of the Association for Computational Linguistics, 10:145–162.

Surangika Ranathunga and Nisansa de Silva. 2022. Some languages are more equal than others: Probing deeper into the linguistic disparity in the nlp world. arXiv preprint arXiv:2210.08523.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In International Conference on Machine Learning, pages 4596–4604. PMLR.

Saleh Soltan, Haidar Khan, and Wael Hamza. 2021. Limitations of knowledge distillation for zero-shot transfer learning. In Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing, pages 22–31, Virtual. Association for Computational Linguistics.

Mukund Srinath, Shomir Wilson, and C Lee Giles. 2021. Privacy at scale: Introducing the PrivaSeer corpus of web privacy policies. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 6829–6839, Online. Association for Computational Linguistics.

Chaofan Tao, Lu Hou, Wei Zhang, Lifeng Shang, Xin Jiang, Qun Liu, Ping Luo, and Ngai Wong. 2022a. Compression of generative pre-trained language models via quantization. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4821–4836, Dublin, Ireland. Association for Computational Linguistics.

Chaofan Tao, Lu Hou, Wei Zhang, Lifeng Shang, Xin Jiang, Qun Liu, Ping Luo, and Ngai Wong. 2022b. Compression of generative pre-trained language models via quantization. arXiv preprint arXiv:2203.10705.

Jörg Tiedemann. 2020. The tatoeba translation challenge – realistic data sets for low resource and multilingual MT. In Proceedings of the Fifth Conference on Machine Translation, pages 1174–1182, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. Advances in neural information processing systems, 30.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2022. Expanding pretrained models to thousands more languages via lexicon-based adaptation. arXiv preprint arXiv:2203.09435.

Hao Wu, Patrick Judd, Xiaojie Zhang, Mikhail Isaev, and Paulius Micikevicius. 2020. Integer quantization for deep learning inference: Principles and empirical evaluation. arXiv preprint arXiv:2004.09602.

Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. 2021. Vocabulary learning via optimal transport for neural machine translation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 7361–7373, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 483–498, Online. Association for Computational Linguistics.

Ziqing Yang, Yiming Cui, Zhigang Chen, and Shijin Wang. 2022. Cross-lingual text classification with multilingual distillation and zero-shot-aware training. arXiv preprint arXiv:2202.13654.

Ali Hadi Zadeh, Isak Edo, Omar Mohamed Awad, and Andreas Moshovos. 2020. Gobo: Quantizing attention-based nlp models for low latency and energy efficient inference. In 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), pages 811–824.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert. In 2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition (EMC2-NIPS), pages 36–39.

Shiyue Zhang, Ben Frey, and Mohit Bansal. 2022. How can NLP help revitalize endangered languages? a case study and roadmap for the Cherokee language. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1529–1541, Dublin, Ireland. Association for Computational Linguistics.

Xinlu Zhang, Xiao Li, Yating Yang, and Rui Dong. 2020. Improving low-resource neural machine translation with teacher-free knowledge distillation. IEEE Access, 8:206638–206645.

A Appendix

A.1 Details of Data Sources

For all the languages in Table 1 we now describe the training and evaluation corpora used. Note that for languages like Assamesse, Odia, Punjabi, etc. we could have accessed a monolingual corpus to supplement our training as well but since we wouldn't have been able to leverage data at a similar scale and quality for the entire language set, we abstained from using methods that leveraged monolingual corpora in these languages.

Bribri Training data from Feldman and Coto-Solano (2020) containing about 7K parallel sentences. Test data from Mager et al. (2021) with 1003 sentences.

Wixarica Training data from Feldman and Coto-Solano (2020) containing about 8k parallel sentences. Test data from Mager et al. (2021) with 1K sentences.

Mundari We requested Indian Institute of Kharagpur for Data on Mundari. This corpus contained 10K parallel sentences. We partition train and test sets from this and generate a test set of 980 sentences [6]

Gondi Data obtained from CGNET (2019) containing 26K sentences. We partition train and test sets from this and generate a test set of 730 sentences6.

Assamesse Train data obtained from Ramesh et al. (2022) containing 0.14 parallel sentences. Test data from (Goyal et al., 2022) containing 1012 sentences

[6] To avoid any test-set leaks, we deduplicate the data by removing tuples (S^i, T^i) where S^i is the i^{th} sentence in the source language and T^i is $i^{th}$the sentence in the target language, between the train and the test set.

Odia Train data obtained from Ramesh et al. (2022) containing 1M parallel sentences. Test set from WAT2021 (Nakazawa et al., 2021) containing 2390 sentences

Punjabi Train data obtained from Ramesh et al. (2022) containing 2.42M parallel sentences. Test set from WAT2021 (Nakazawa et al., 2021) containing 2390 sentences

Gujarati Train data obtained from Ramesh et al. (2022) containing 3.05M parallel sentences. Test set from WAT2021 (Nakazawa et al., 2021) containing 2390 sentences

A.2 Evaluating Continued Pretaining with Synthetically Augmented or Lexicon Adapted Monolingual Data for improving the HM

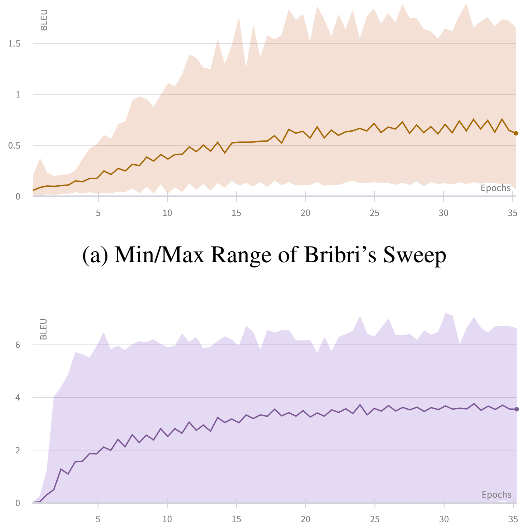
The use of continual pretraining with monolingual data has been shown to be very useful in improving the transfer for low-resource languages. In our cases, our lowest resource languages, i.e, Bribri, Wixarica, Gondi and Mundari, did not have any monolingual data available natively so we explored the augmentation of the same using lexicons (Wang et al., 2022). We also generated forward translated data using the HM that we developed to fuse with the lexicon-adapted data. For continued pretraining we use a fixed learning rate of 0.001. Results of our experiments are logged in Table 5.We use the following notations to report our results GMD- *Gold Monolingual Data,* LA- *Lexicon Adapted Monolingual Data,* KDD- *Knowledge Distilled Monolingual Data* where GMD indicates the target-side monolingual data available within the parallel corpus of the language, KDD indicates the forward-translated data that we generate via our best-performing model for Gondi i.e., mt5-base. We generated 100K labels using mt5-base teacher, and also experimented adding 100K sentences from a weaker teacher, i.e., mt5-small in hopes of leveraging a more diverse class of labels to train the student on.

We did observe a small gain in performance upon the addition of LA data during pretraining - though the post-quantization performance and the distilled model' significant performance degradation called for a deeper investigation on the effects of continued pretraining for this language.

A.3 Hyperparameter Trial Configurations

We ran Hyperparameter sweeps with the configurations specified in Table 5.

Note that in congruence with the observations of subsection 4.6, we also provide the min-max range of performance for Gondi and Bribri in Figure 6.



(a) Min/Max Range of Bribri's Sweep

(b) Min/Max Range of Gondi's Sweep

Figure 6: Variation of performance across languages

As can be observed, for a set of hyperparameters, at least one of which is optimal for some other language in the set, both languages fail to converge. Similarly, in extension to subsection 4.4, we also checked if for the same hyperparameter set, the variation in student architecture produced significant performance variations.

The results demonstrated in Figure 7 did not show any significant variation except for the case of Gondi, i.e., altering the student architecture - while keeping all other priors the same: adversely affected the performance in that one case.

A.4 Comparing Size-Reduction Affinity of Quantization and Distillation

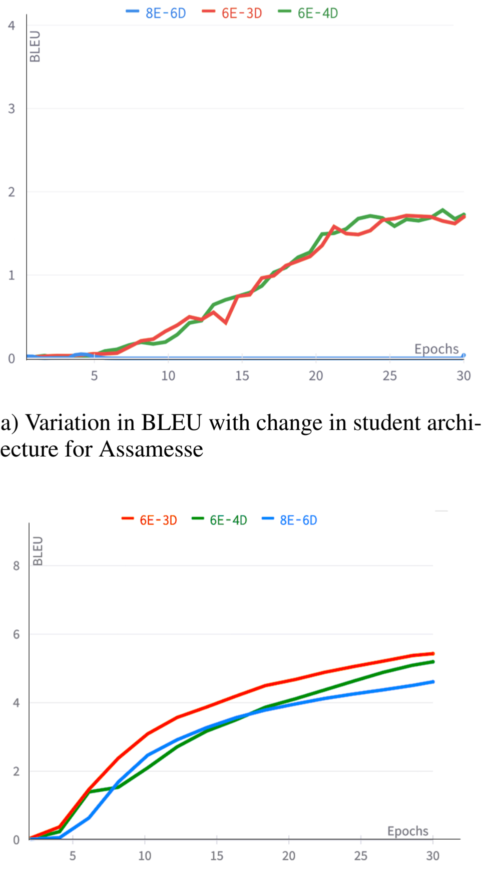
This exploration is extremely useful as the size of a model significantly impacts several factors associated with the consumption of any service, impacting it's adoption by community members through several ways including *(a) Accessibility on Edge:* Since mobile devices are constrained in their RAM and Memory Usage - users with edge devices of low-capabilities are naturally inhibited to is services that drain their device's resources. *Inadequate Connectivity Requirement for Inference, One-time download and Service Updates:* Users

Model	Data	spBLEU	S(M) (in MB)
Transformer	$26.2k$	1.4	240
mT5-small	$61.9k$	12.7	1200
mT5-small	$26.2k$	14.3	1200
mT5-base	$26.2k$	15.6	2100
mBART	$26.2k$	13	2280
mT5-small: CPT {GMD }	$26.2k^{mono}$	14.9	1200
mT5-small: CPT {LA }	$200k^{mono}$	**14.9**	1200
mT5-small: CPT {LA }	$200k^{mono}$	**10.8**	400
mT5-small: CPT {KDD }	$143k^{mono}$	15.2	1200
mT5-small: CPT {GMD + LA + KDD }	$26.2k + 343k^{mono}$	14.7	1200
mT5-small: Quantizing M1	$26.2k$	13.8	400
Quantizing CPT Model {Best mT5-small }	$26.2k$	10.2	400
Transformer + KD	$26.2k + 240k$	10.1	185

Table 4: Gondi: Use of Lexicon Adaptation, Continued Pretraining and Mixed-training with Lexicon Adapted and Forward Translated Monolingual Data.

Hyperparameter	Candidate Values
Train batch size	32, 64
Epochs	10, 30, 60
Method	grid
Metric	BLEU
Gradient Accumulation	2, 4
Label Smoothing	0, 0.1
Learning Rate	5{e-5,e-5,e-6}
Warmup Steps	500, 1000

Table 5: Candidate values of hyperparameters: Sweep for finding the optimal hyperparameter set for Distillation



(a) Variation in BLEU with change in student architecture for Assamesse

(b) Variation in BLEU with change in student architecture for Gondi

Figure 7: In the legend E and D refers to Encoders and Decoders respectively

may often avoid downloading apps that seem too large, particularly in emerging markets where devices connect to often-spotty 2G and 3G networks or work on pay-by-the-byte plans [7]. *Large Rendering Time:* Finally, a bloated size may often be associated with a larger rendering response period which might hinder the usability experience of a user engaging with the MT service.

Note on Inference Times In theory, compression through both distillation and quantization is expected to be conducive to faster inference for the models: The distilled models are not bounded to use a pretrained embedding and hence can gain in inference by using smaller, target-language specific embeddings. The quantized models can also benefit due to the reduced precision in which the inference operations are carried out, though this optimization is heavily dependent on if the hardware running the model can leverage these operations in

[7] https://developer.android.com/topic/performance/reduce-apk-size

Language	Native S(HM)	Compressed S(Q,D)
Bribri	1228	(400, 153)
Wixarica	1228	(400, 153)
Gondi	1228	(400, 153)
Mundari	1228	(400, 153)
Assamesse	1228	(400, 189)
Odia	1228	(400, 189)
Punjabi	232	(75, 189)
Gujarati	232	(75, 189)

Table 6: Sizes of the Uncompressed and Compressed Variants for all languages - Q and D indicate the compressed sizes of the Quantized and the Distilled Models respectively. All sizes are in MB.

their expected precision (Bondarenko et al., 2021). Especially in the case of quantization, the scope of this analysis would be quite vast, which is why we also excluded it from our current analysis.

Data Augmentation for Inline Tag-Aware Neural Machine Translation

Yonghyun Ryu, Yoonjung Choi, Sangha Kim
Samsung Research
Seoul, Republic of Korea
{yonghyun.ryu, yj0807.choi, sangha01.kim}@samsung.com

Abstract

Despite the wide use of inline formatting, not much has been studied on translating sentences with inline formatted tags. The detag-and-project approach using word alignments is one solution to translating a tagged sentence. However, the method has a limitation: tag reinsertion is not considered in the translation process. Another solution is to use an end-to-end model which takes text with inline tags as inputs and translates them into a tagged sentence. This approach can alleviate the problems of the aforementioned method, but there is no sufficient parallel corpus dedicated to such a task. To solve this problem, an automatic data augmentation method by tag injection is suggested, but it is computationally expensive and augmentation is limited since the model is based on isolated translation for all fragments. In this paper, we propose an efficient and effective tag augmentation method based on word alignment. Our experiments show that our approach outperforms the detag-and-project methods. We also introduce a metric to evaluate the placement of tags and show that the suggested metric is reasonable for our task. We further analyze the effectiveness of each implementation detail.

1 Introduction

While most machine translation studies are focused on plain text, the textual information that we encounter every day on the internet contains words with different styles and links within the sentence. Various styling of any part of the text is called inline formatting and is represented by markup and markdown tags. The inline formatting not only improves the readability of documents but also provides additional information with tags; so it is important to correctly translate sentences including tag information. In addition, the widespread use of formatting tags in the computer-based document system makes it inevitable to increase the demand for translating web text or structured documents containing inline tags.

There are two main approaches to translating segments with inline tags. One solution is the detag-and project method (Hanneman and Dinu, 2020). It first strips tags from the source sentence and translates only the plain text. Then, the removed tags are reinserted into the translation results using word alignments, which can be induced from attention weight in the model or an external aligner such as SimAlign (Jalili Sabet et al., 2020). This method does not take into account the re-insertion of tags in the translation process, making it difficult to restore tags at the proper positions.

Another way is to use an end-to-end model which takes sentences including tags as inputs and generates translation results with tags. Since tag information is considered, the translation can be performed with more context, thus this method potentially improves the quality of translation and the placement of tags. To train end-to-end models, a parallel corpus, where both source target sentences contain aligned tags, is required. Even though a parallel corpus with markup tags was released by Hashimoto et al. (2019), their data is limited to the domain of online help and there is still not many of such data available to train a high-quality model.

To address this lack of tagged parallel corpus, Hanneman and Dinu (2020) introduces a data augmentation approach using tag injection. Their method is to insert tags into corresponding fragments in the source and the target. In their approach, the aligned phrases are identified by an exhaustive search by matching all translated source fragments with all target fragments. This method has two drawbacks by its nature. The first is that their approach requires a high computational cost because it requires computing translation for all possible phrases for at least millions of parallel sentences to train a model. Secondly, only constrained tags can be augmented because they find corresponding pairs with out-of-context translation.

In this paper, we propose an efficient and effec-

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 886 - 894
December 7-8, 2022 ©2022 Association for Computational Linguistics

tive tag augmentation method using word alignments (Brown et al., 1993) to overcome the above shortcomings. Our method uses an external word aligner to compute correspondence between the source and target words, and find aligned fragments by phrase extraction algorithm (Och et al., 1999). Then tags are inserted according to the phrasal alignments. The tag-augmented parallel corpus by this method can train a model that translates sentence containing tags in an end-to-end way.

For comparisons, we implement competitive baselines and propose a metric to automatically evaluate the placement of tags. Through experiments, we show that our approach is superior to the detag-and-project methods and demonstrate the effectiveness of each implementation detail.

2 Method

In this section, we propose an efficient and effective method to insert inline tags into an existing parallel corpus. In augmented data, the position of tags in the source segment must be preserved in the target segment. The word "*preserved*" means that tags in the target sentence must surround spans with the same role and meaning as the corresponding source spans. In other words, the source and target fragment in the same tag has to correspond with each other. Moreover, inline tags can contain not only a word but also a phrase or even any consecutive words. Therefore, how to find corresponding phrase[1] pairs for each parallel sentence is the key to synthesizing tag-aligned parallel data. This makes our method focus on finding aligned phrase pairs.

Our proposed augmentation method consists of three steps. We first generate word alignments for the parallel corpus using external word aligners (Section 2.1). Then we extract aligned phrase pairs for each sentence pair with the word alignments (Section 2.2). Lastly, for each parallel sentence and the aligned phrase pairs, since each sentence usually has a lot more aligned pairs than the number of words in the sentence, we randomly select some of the pairs and insert tags to surround the phrases (Section 2.3). Figure 1 presents the whole process of our methods. The example is from Philipp Koehn's lecture[2].

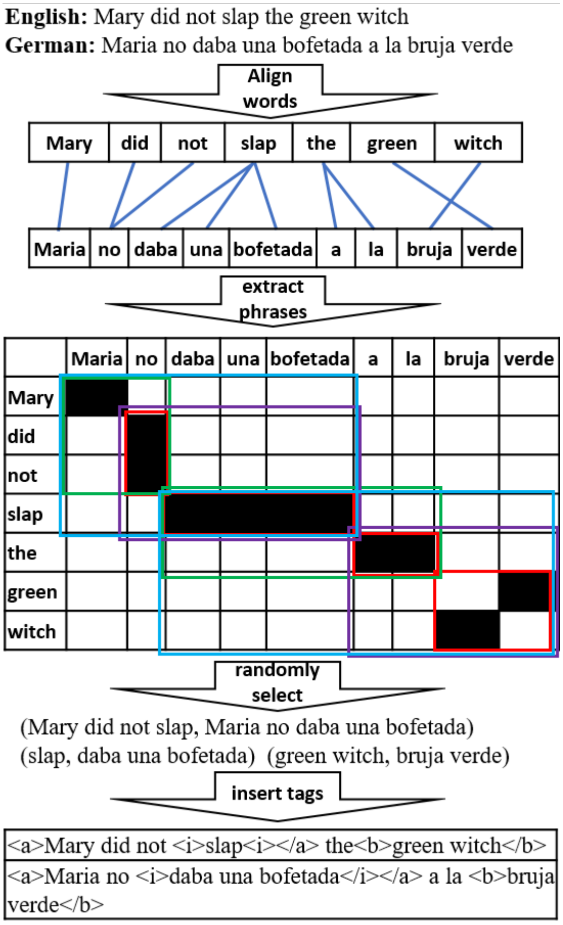

English: Mary did not slap the green witch
German: Maria no daba una bofetada a la bruja verde
Align words
Mary
did
not
slap
the
green
witch
Maria
no
daba
una
bofetada
a
la
bruja
verde
extract phrases
randomly select
(Mary did not slap, Maria no daba una bofetada)
(slap, daba una bofetada) (green witch, bruja verde)
insert tags
<a>Mary did not <i>slap<i></a> the<b>green witch</b>
<a>Maria no <i>daba una bofetada</i></a> a la <b>bruja verde</b>

Figure 1: The process of our methods.

2.1 Word Alignment

Word alignment represents word-level correspondence in a parallel sentence. In statistical machine translation, implementation of IBM models (Brown et al., 1993) such as FastAlign (Dyer et al., 2013) and GIZA++ (Och and Ney, 2003) are famous to compute word alignment from parallel corpus. As deep neural network-based aligners, there are SimAlign (Jalili Sabet et al., 2020) and AwesomeAlign (Dou and Neubig, 2021). These neural methods use the similarity of contextual embeddings based on pretrained multilingual models to compute the correspondence between source and target words.

Our approach starts by using one of the above external word aligners to compute forward (source-to-target) and backward (target-to-source) word alignment, and then apply symmetrizing heuristics (Koehn et al., 2005) such as `grow-diag` and `grow-diag-final-and` for better alignment.

[1] In this paper, the word "*phrase*" indicates consecutive words of any length. The length can be 1 and more.

[2] https://wiki.eecs.yorku.ca/course_archive/2014-15/W/6339/_media/esslli-slides-day3.pdf

2.2 Phrase Extraction

The phrase level alignment has been proposed in Och et al. (1999) to improve statistical machine translation.

We find aligned phrase pairs depending on word alignments with phrase extraction algorithms. The phrase extraction algorithm finds phrasal alignments by exhaustively searching all phrase pairs that are consistent with word alignment. For the detailed algorithm description, please see the NLTK implementation (Loper and Bird, 2002)[3].

We do not use phrase probability tables (Koehn et al., 2003) to refine aligned pairs since it prevents the collection of diverse kinds of phrases. Instead, we do not allow phrases with unaligned words for more accurate phrase extraction[4].

2.3 Tag Insertion

In this step, we insert tags that surround aligned phrase pairs. The number of tags is randomly selected to less than 30% of the number of words. Then the aligned phrase pairs are randomly chosen as many as the number of tags, and tags are inserted according to the pairs. In this process, we can insert tags following the HTML syntax, and the tags can be nested.

2.4 Augmentation Cost Analysis

The cost of augmentation is crucial since machine translation models typically are trained on more than millions of parallel sentences. For this reason, we roughly analyze the amount of computation to show that our method is cost-efficient over the previous approach.

The previous tag augmentation method suggested in Hanneman and Dinu (2020) uses a machine translation model to find corresponding fragments with exhaustive search. Their method needs at least $O(n * m)$ translation model inferences for each parallel sentence, where n is the size of the maximum corresponding phrase length and m is the number of tokens in the source.

Assume that our method use SimAlign (Jalili Sabet et al., 2020) for a word aligner. Our approach requires one XLM-R (Conneau et al., 2020) inference to compute contextual word embeddings and single matrix multiplication to get cosine similarities between tokens. Since single XLM-R inference cost much to single matrix multiplication, we can count single XLM-R inference as the amount of computation. Alignment symmetrizing heuristic algorithms and phrase extraction are also required for our approach, but we do not count it to time comparison since these algorithms also take much less time than neural model inference.

Because the translation model uses beam search, both XLM-R and a translation model have almost the same computation cost, but the translation has more latency because it is autoregressive. In short, the previous method needs $O(n * m)$ translation model inference but our approach only requires a single XLM-R inference for each sentence pair. Therefore, our model is more efficient than translation-based augmentation.

3 Experimental Setup

3.1 Data

3.1.1 Training Data

Our data augmentation goal is to train an end-to-end model to translate inline tagged text with competitive translation quality. For a fair comparison of translation performance, we use the same training and test sets as Edunov et al. (2018) and Garg et al. (2019). The tagged parallel corpus released by Hashimoto et al. (2019) is also used to evaluate the placement of tags.

WMT'18 This dataset is a set of parallel corpora for the WMT'18 English-German news translation task (Bojar et al., 2018) and consists of the Europarl v7, common crawl, news commentary v13, and rapid corpus of EU press releases. Parallel sentences with either a sentence longer than 250 tokens or a source/target token length ratio exceeding 1.5 are removed[5].

LXM In this paper, we call the dataset released by Hashimoto et al. (2019) **LXM** which is their GitHub repository name's initials[6]. The data have parallel sentences with aligned inline tags. For German-to-English, there are about 100,000 train pairs and 2,000 development pairs. Only about a quarter of them contain tags.

[3] https://www.nltk.org/_modules/nltk/translate/phrase_based.html

[4] The original phrase extraction implemented in the NLTK includes unaligned words in the aligned phrase since it is still considered to be consistent with word alignment.

[5] We use the XLM-R tokenizer to filter the parallel corpus.

[6] https://github.com/salesforce/localization-xml-mt

LXM-plain This data is the **LXM** training data without tagged pairs. Since the domain of LXM is online help and WMT'18 corpora do not cover them, thus we add this data to training data. In this paper, we prove the effectiveness of the tag augmentation approach, thus we only use plain sentences from the training set.

3.1.2 Test Data

For comparison of our approach to the previous works, we use newstest2014 (**WMT'14**) to evaluate translation quality and LXM development set (**LXM-dev**) for the accuracy of tag placement.

3.2 Naive End-to-end Baseline

This baseline takes text with markup tags as inputs and handles them like plain text but uses a model which has been trained without tagged parallel corpus.

3.3 Detag-and-project Baselines

The detag-and-project approach (Hanneman and Dinu, 2020) first strips tags from the source sentence, translates the plain one, and then places the removed tags in the corresponding positions according to the word alignments. During the projection stage, one tag can be projected into separated parts, in this case, we insert one minimum-sized tag that surrounds all of the parts, which is also called the Min-Max Tag Pair Projection in Zenkel et al. (2021).

We establish three detag-and-project baselines according to the way to get word alignment.

Layer Average Baseline The layer average baseline is to extract word alignments from the attention. There are two methods to induce word alignments from the attention (Chen et al., 2020): NAIVE-ATT and SHIFT-ATT. Word alignments are induced from attention scores between the encoder and decoder. NAIVE-ATT (Garg et al., 2019) relates the maximum attention scores with the decoder's output token and uses attention weight of the penultimate layer of the decoder. SHIFT-ATT (Chen et al., 2020) associates the maximum attention scores with the decoder's input token and uses attention weight of the third layer of the decoder.

Garg et al. (2019) This baseline can be simply called attention enhanced approach. Like the layer average baseline, this method also extracts word alignments from attention scores, but it uses the trained attention head by multi-task learning. Specifically, one attention head of the fifth layer of the decode is jointly trained with translation by word alignments from GIZA++ (Och and Ney, 2003). Furthermore, full target context is used when the attention weight learns word alignments and predicts alignments from cross-attention. We re-implement the model by the author's Fairseq (Ott et al., 2019) implementation[7] to reproduce their results. In this paper, we do not apply any pre-tokenizer, and only use an unigram language model tokenizer (Kudo, 2018) with a vocabulary size of 35,000. All other hyperparameters are the same as Garg et al. (2019).

SimAlign This method uses SimAlign (Jalili Sabet et al., 2020) as an external aligner to restore tags on the translation results. For this model, the layer average baseline model is used to generate translated sentences. We take **argmax**[8] function to extract each direction of word alignment and apply `grow-diag-final-and` heuristics (Koehn et al., 2005) to symmetrize the bidirectional alignments for better word alignments.

3.4 Implementation Details

Reversible Tokenization The previous works use Moses tokenizer (Koehn et al., 2007) as a pre-tokenizer before applying Byte-Pair-Encoding (Sennrich et al., 2016). However, we don't use any pre-tokenizers like Moses because it is impossible to detokenize tokenized results to the original sentence completely even if a well-designed rule-based detokenizer is applied. We only apply SentencePiece (Kudo and Richardson, 2018) for tokenization, because it is a reversible tokenizer and makes a purely end-to-end system possible. A unigram language model tokenizer (Kudo, 2018) is trained from the WMT'18 corpus only without applying subword regularization.

Whitespace Shift As SentencePiece (Kudo and Richardson, 2018) tokenizer adds dummy whitespace at the beginning of a sentence, we move the whitespace before the tag to the back of the tag since the whitespace at the beginning of a word plays an important role in tokenization because the whitespace is also considered a target to tokenize by the subword tokenizer. A word without whitespace at the beginning is often tokenized differently from a word with a whitespace. For example, "World" is

[7] `https://github.com/facebookresearch/fairseq/tree/main/examples/joint_alignment_translation`

[8] Argmax aligns words to the most similar word.

	WMT'14	LXM-dev				
Model	**BLEU**	**BLEU**	**XML BLEU**	**XML Acc.**	**XML Match**	**F1**
Edunov et al. (2018)	29.0					
Hashimoto et al. (2019)		52.91	51.16	99.75	99.3	
Naive End-to-end						
- WMT'18 only	28.7	25.12	22.14	98.05	95.15	44.83
- WMT'18 + LXM-plain	28.8	51.05	49.9	98.6	98.2	58.93
Layer Average Baseline						
- NAIVE-ATT	28.8	52.22	50.45	100	98.5	60.53
- SHIFT-ATT	28.8	52.22	50.71	100	98.75	61.59
Garg et al. (2019)	28.7	52.46	50.64	100	98.0	68.04
SimAlign	28.8	52.22	48.43	100	97.55	60.96
Tag Augmentation (ours)	29.1	53.37	52.8	100	99.35	74.31
Tag Shift	29.1	53.37	52.07	100	99.35	53.75

Table 1: Evaluation results on the WMT'14 and LXM-dev. Models are trained with WMT'18 and LXM-plain by default. In *Tag Shift*, all tags are moved to one word to the left in the translation hypothesis.

tokenized into "Wo", "r", "ld", however, "_World" is tokenized into "_Wor", "ld". This inconsistency affects adversely translation quality. For this reason, we move the space and put it back in the pre- and post-processing step.

Tag Replacement Markup tags often have attributes and the attributes generally don't need to be translated and just copied to the translation. Like other approaches (Müller, 2017) and (Hanneman and Dinu, 2020), we replace the real tags with indexed special tags. We insert at most 9 tags per each parallel pair in tag augmentation. In our implementation, we use "<a_0>,<a_1>,...<a_9>,</a_0>,</a_1>,...</a_9>" as special tokens. In the training step, the index of special tokens is shuffled for training efficiency. In the inference, we convert real tags to the special tokens and the convert table in the pre-processing step. After translation, we revert them to the original tags in post-processing.

Tag Augmentation Hyperparmeters We use subword alignments instead of word alignments since according to recent studies (Garg et al., 2019) (Jalili Sabet et al., 2020), and (Dou and Neubig, 2021); subword-based alignments outperform word alignments on AER (Alignment Error Rate). Since our SimAlign baseline uses the XLM-R model (Conneau et al., 2020), for a fair comparsion, parallel corpus is tokenized by the XLM-R tokenizer[9] before computing word alignment with statistical models.

Like Garg et al. (2019), for our augmentation, Giza++ with 5 iterations of IBM1, HMM, IBM3 and IBM4 are used as a word aligner. However, for training an end-to-end model, we use a different tokenizer as explained in 3.4. For end-to-end training, we use the combination of tagged data and plain data in a 1:1 ratio.

Model Training Hyperparameters Basically for all experiments, we follow the same hyperparameters as the Align and Translate Task of Garg et al. (2019). We use the fairseq toolkit (Ott et al., 2019) for all of our experiments. The big transformer architecture[10] with the post layer normalization is used for all experiments. The difference is that we use learning rate of 5e-4, learning rate warmup over the first 8000 steps, and a batch size of 32768 tokens[11] on 8 A100 GPUs for 120k updates. We use the checkpoint which averages the last 10 checkpoints, and a beam size of 5 for inference.

4 Evaluation

In this section, we describe several metrics to evaluate our methods and present experimental results.

4.1 Evaluation Metrics

For comparison of translation quality to Garg et al. (2019) and Edunov et al. (2018), we use

[9] They use SentencePiece tokenizer and the model can download in `https://github.com/facebookresearch/XLM`.

[10] The architecture name we used in faisreq is `trasnformer_wmt_en_de_big`.

[11] Actually, we use 16384 tokens with accumulating 2 update gradients.

	WMT'14	LXM-dev			
Variation	**BLEU**	**BLEU**	**XML BLEU**	**XML Match (Acc.)**	**F1**
Baseline	29.1	53.37	52.8	99.35	74.31
w/o Tag Replacement					
trained on plain corpus	28.8	51.05	49.9	98.2 (98.6)	58.93
trained on tagged corpus	28.7	52.44	51.62	99.05 (99.85)	71.14
Symmetric Heuristics					
`intersection`	29.2	53.13	52.4	99.2	60.13
`grow`	28.9	53.21	52.52	99.2	74.73
`grow-diag-final-and`	29.0	53.11	52.39	99.2	75.35
Phrase Length					
8	28.8	53.33	51.9	99.05	72.07
16	28.9	53.17	52.86	99.45	73.61
32	29.1	52.85	52.25	99.25	73.12
128	28.9	52.92	52.55	99.45	72.84
Word Aligner					
Fast-Align	29.0	52.89	52.14	99.2	72.99
SimAlign	28.9	52.99	52.41	99.15	73.5
w/o whitespace shift	28.6	52.88	52.26	99.4	71.77
NLTK phrase extraction	28.9	52.85	51.96	99.3	72.48
Violating HTML syntax	29.1	53.09	52.29	99.2	72.58
Tagged data only	28.5	52.59	51.66	99.35	72.27

Table 2: Variations on tag augmentation. The baseline uses `grow-diag` heuristics, phrase length of 64, GIZA++ as a word aligner, and improved phrase extraction. The score on XML Accuracy is not mentioned because all scores are 100. *w/o whitespace shift* do not apply whitespace shift in the training and inference.

sacreBLEU (Post, 2018) with **WMT'14**. Like Hashimoto et al. (2019)'s work, we use BLEU, XML BLEU, XML Accuracy, and XML Match as metrics in the evaluation of LXM-dev. We also use the same evaluation scripts as they do[12]. For accurate evaluation of the tag placement, we introduce an F1 score-based metric to evaluate the position of tags by focusing on the words that tags surround.

BLEU and XML BLEU The BLEU score here is the same as the existing BLEU score measured in plain text. For that, all tags first are removed if exist, and then the BLEU is measured using the same tokenizer as Hashimoto et al. (2019). The XML BLEU uses the same metric, but if there are tags, the BLEU score is measured with text containing XML tags. The XML tags are also considered to compute the score.

XML Accuracy and Match The XML accuracy is the ratio of the valid XML outputs in all translation results. The XML match is the ratio of the outputs that have the same XML structure as the reference.

F1 score This metric is introduced to evaluate the placement of tags. Since the goal of tag transfer is to surround the corresponding words accurately, we introduce a metric to focus on evaluating words surrounded by tags. In this sense, we make use of the metric from SQuAD (Rajpurkar et al., 2016), since it evaluates the words in the span. SQuAD's answers consist of a span of consecutive words in a paragraph and they evaluate how accurately the span contains the correct answer. Since what we really want to evaluate is not the position of tags but the content of the span surrounded by tags, the goal of their evaluation is similar to ours in that they aim to assess a range of words.

We apply this metric to evaluate the accuracy of tag placement. The score measures the overlap between the ground truth and the prediction to calculate a score. More precisely, they treat the hypothesis and the reference as bags of words, and calculate F1. Unlike SQuAD dataset, LXM-dev can have more than one tag for each sentence, thus

[12] `https://github.com/salesforce/localization-xml-mt`

Model	Alignment Error Rate (AER)					
Method	**SHIFT-ATT**			**NAIVE-ATT**		
Layer	**1**	**2**	**3**	**4**	**5**	**6**
Layer Average Baseline	**29.1**	31.8	36.1	41.9	42.8	51.2
Tag Augmentation (ours)	27.9	**26.4**	49.6	37.8	35.7	44.7
Garg et al. (2019) (all heads)	32.0	26.0	22.7	35.3	29.1 (**20.5**)*	72.2

Table 3: Results on Vilar et al. (2006). * uses the first head trained by word alignments. Others use the average. While we apply SHIFT-ATT for the half bottom layers, we apply NAIVE-ATT on the top 3 layers for better performance on AER (Chen et al., 2020).

we use the average score per tag[13].

4.2 Results

Firstly, in order to show the relevance of the proposed F1 metric, we shift all tags to the left by one word, which must cause performance degradation in tag placement. In the results of *Tag Shift* in Table 1, compared to *Tag Augmentation*, there is only a slight drop on XML BLEU, however, the F1 score shows a significant decrease. This implies that the proposed metric is reasonable to evaluate the placement of tags.

In Table 1, there are the results of the baselines and our tag augmentation method. The experimental results show that our augmentation method achieves the best performance for all metrics. Furthermore, according to the XML Accuracy, the tag augmentation model is able to generate all XML tags grammatically correctly in the source without XML-constrained beam search.

4.3 Augmentation Variation

We conduct various experiments to figure out what greatly affects the performance. Firstly, we note that tag augmentation with intersection heuristics causes considerable degradation on the f1 score. We also note that according to (Dou and Neubig, 2021), the performance of word alignments between Fast-Align and Giza++ is considerable, but the models trained on each data show relatively similar performance compared to the AER scores.

Even though there is a little gap in performance, the result indicates that all of our proposed implementation details have a positive influence on both translation quality and tag placement. As a result, performance improvement is achieved by all factors combined.

4.4 Indirect Learning Alignment

We further investigate the effect of the aligned tagged corpus. Table 3 shows that the AER score from all layers is improved than the *Layer Average Baseline*, but does not reach the score of multi-task training model (Garg et al., 2019) where word alignments are trained directly. This result indicates that the tag-augmented data help models' attention to learn the correspondence between source and target words indirectly.

5 Conclusion

In this paper, we have presented an efficient and effective inline tag augmentation method to insert tags into existing parallel corpora using a word aligner and the phrase extraction algorithm. Our approach injects inline tags economically and accurately.

We also introduced a reasonable metric for the automatic and accurate evaluation of the placement of tags and analyzed the effectiveness of the detailed methods used in our approach. The experiment results show that the model trained on data augmented by our method outperforms the previous detag-and-project methods.

References

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter

[13] Unfortunately, some sentences in LXM-dev have multiple of the same name tags in a sentence. Because there is no way to align the same name tags, we regard the multiple separate spans with the same name as one consecutive span in the evaluation.

estimation. *Computational Linguistics*, 19(2):263–311.

Yun Chen, Yang Liu, Guanhua Chen, Xin Jiang, and Qun Liu. 2020. Accurate word alignment induction from neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 566–576, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China. Association for Computational Linguistics.

Greg Hanneman and Georgiana Dinu. 2020. How should markup tags be translated? In *Proceedings of the Fifth Conference on Machine Translation*, pages 1160–1173, Online. Association for Computational Linguistics.

Kazuma Hashimoto, Raffaella Buschiazzo, James Bradbury, Teresa Marshall, Richard Socher, and Caiming Xiong. 2019. A high-quality multilingual dataset for structured documentation translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 116–127, Florence, Italy. Association for Computational Linguistics.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of the Second International Workshop on Spoken Language Translation*, Pittsburgh, Pennsylvania, USA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Mathias Müller. 2017. Treatment of markup in statistical machine translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 36–46, Copenhagen, Denmark. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och, Christoph Tillmann, and Hermann Ney. 1999. Improved alignment models for statistical machine translation. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

David Vilar, Maja Popovic, and Hermann Ney. 2006. AER: do we need to "improve" our alignments? In *Proceedings of the Third International Workshop on Spoken Language Translation: Papers*, Kyoto, Japan.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2021. Automatic bilingual markup transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3524–3533, Punta Cana, Dominican Republic. Association for Computational Linguistics.

The SPECTRANS System Description for the WMT22 Biomedical Task

Nicolas Ballier[1], Jean-Baptiste Yunès[2], Guillaume Wisniewski[3],
Lichao Zhu[3], Maria Zimina-Poirot[1]

[1]CLILLAC-ARP, [2]IRIF, [3]LLF
Université Paris Cité, F-75013 Paris, France
{nicolas.ballier, guillaume.wisniewski, jean-baptiste.yunes,
lichao.zhu, maria.zimina-poirot}@u-paris.fr

Abstract

This paper describes the SPECTRANS submission for the WMT 2022 biomedical shared task. We present the results of our experiments using the training corpora and the JoeyNMT (Kreutzer et al., 2019) and SYSTRAN Pure Neural Server/ Advanced Model Studio toolkits for the language directions English to French and French to English. We compare the predictions of the different toolkits. We also use JoeyNMT to fine-tune the model with a selection of texts from WMT, Khresmoi and UFAL data sets. We report our results and assess the respective merits of the different translated texts.

1 Introduction

For this WMT22 Biomedical workshop, we focused on the selection of texts used for fine-tuning. We selected what we believe to be the two best models we produced for the EN-FR track with two different neural toolkits but we mostly took the opportunity to discuss the translated texts. The rest of the paper is organised as follows: Section 2 summarises our approaches to the task, Section 3 details the training data of our experiments, Section 4 presents the results. Section 5 discusses them.

2 Our Approaches to the Task

This section presents our various strategies for this task and our four submissions. We compared the predictions of two toolkits but our comparison is very partial as the training data differs. We trained several systems with JoeyNMT (Kreutzer et al., 2019) training and fine-tuning with UFAL, WMT and Khresmoi data. We used the SYSTRAN Pure Neural® Server generic system and tried to fine-tune with specialised terminology. We used SYSTRAN Advanced Model Studio® to fine-tune a generic model with in-house data based on 2,700 aligned segments collected during the translation of the French federation for diabetes.[1] Table 1 summarises our submissions.

With JoeyNMT, we selected the training data, comparing the performance with and without the added data and applied fine-tuning to the model based on UFAL medical corpora The following section details the model selection and fine-tuning.

3 Data and Tools Used

In this section, we present different approaches that we adopted to train baseline models and proceed to fine-tuning. We have built two baseline models : one trained with generic data set fine-tuned with in-domain data, and the other trained directly with in-domain data, in order to compare their performances and to better understand functioning of in-domain NMT training.

3.1 Data for baseline models training

We used two baseline models : the first one is built based on our model submitted for WMT 2021. It took the Europarl 7 parallel corpus as data set trained with 341,554 sentences in two directions (EN⇔FR)(Ballier et al., 2021); the second one has been built by using bilingual (EN-FR) in-domain parallel corpora data set UFAL provided by WMT 2022 (with 2,693,509 sentences). The corpora have been normalized and sentences longer that 50 words have been removed. Thus we have retained 2,159,307 sentences. These sentences are split in the ratio of 6-2-2 : 60% for training, 20% for development and the last 20% for evaluation. Two tokenizations are applied to all the data sets : standard tokenization (`Spacy`) segments data into words and BPE tokenization into sub-words with `SentencePiece` (Kudo, 2018).

[1] https://www.federationdesdiabetiques.org. Diabetes terminology proved to be not so useful for the actual test set.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 895 - 900
December 7-8, 2022 ©2022 Association for Computational Linguistics

run	BLEU (into English)	BLEU (from English)	toolkit	training data
run1	0.2581	0.2068	JoeyNMT	baseline with UFAL
run2	0.4010	0.31636	Pure Neural Server	general training data
run3	0.2587	0.0732	ModelStudio Light	fine-tuning with in-house data
run4	0.0969	0.2034	JoeyNMT	UFAL fine-tuned

Table 1: Summary of our official submissions

3.2 Data for fine-tuning

We used two data sets to fine-tune the generic baseline model. For the first data set, we have compiled the WMT Medline parallel corpus since 2016 [2] as well as Khresmoi dev and test data (EN-FR) [3]. The whole data set contains 109,912 sentences. For the second one, we used the normalized and sub-tokenized UFAL data set mentioned above.

4 Experiments and Results

In our experiments, we aimed to compare the different JoeyNMT models (baseline and fine-tuning) that we have trained with SYSTRAN model. JoeyNMT, which is based on TRANSFORMER (Vaswani et al., 2017), requires lighter implementations than OpenNMT (Klein et al., 2017).

4.1 Baseline with JoeyNMT

We have trained a baseline model with in-domain data set UFAL. For FR→EN model, the best checkpoint is recorded at step 60,000 with a BLEU score of 61.01 (PPL: 1.53); as for EN→FR model, the best checkpoint is recorded at step 40,000 with a BLEU score of 59.23 (PPL: 1.45, see Figure 1) [4].

4.2 Fine-tuning with JoeyNMT

The generic baseline model was fine-tuned with the following parameters: vocabulary size: 32,000, maximum sentence length: 50, maximum output length: 100, training initializer: XAVIER, number of layers: 6, number of heads: 8 normalization: tokens, encoder embedding dimension: 512, decoder embedding dimension: 512, hidden size: 512. It was fine-tuned with two data sets. The first one with Medline-Khresmoi data set got the best BLEU score from French to English 54.8, 38.4 as from English to French (see Figure 2).

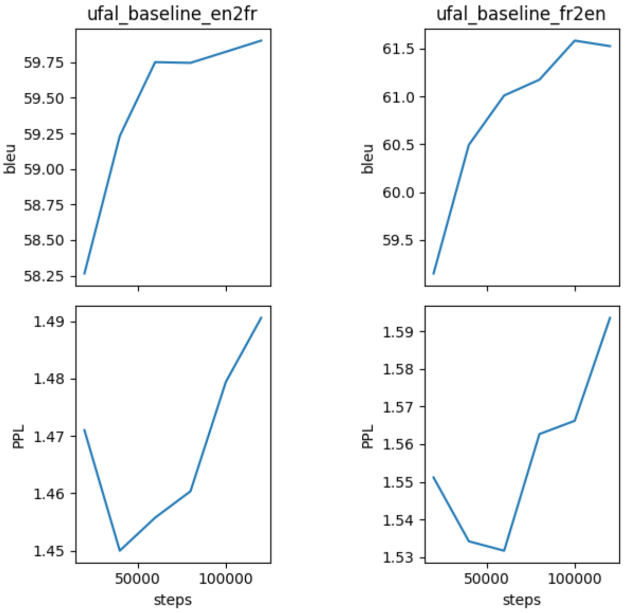


Figure 1: Baseline trained with UFAL data set FR⇔EN

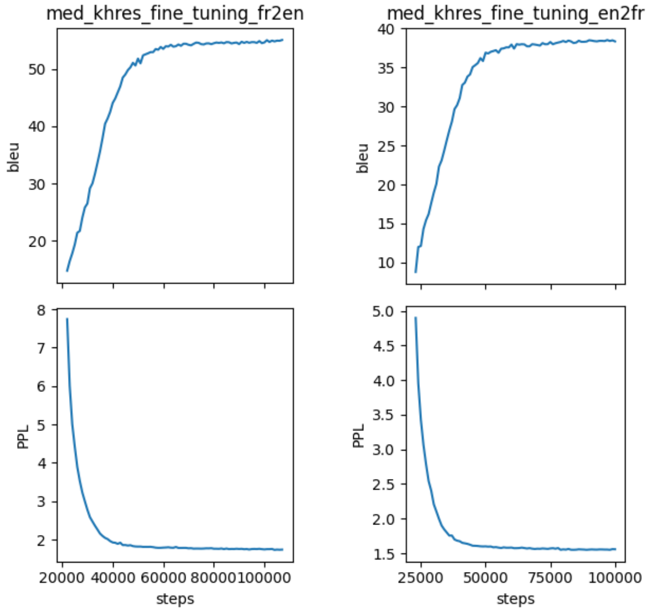


Figure 2: Fine-tuning with Medline and Khresmoi data set FR⇔EN

[2] https://github.com/biomedical-translation-corpora/corpora

[3] http://hdl.handle.net/11234/1-2122

[4] We noticed that the validation processes were extremely long. Every validation after 20,000 steps took about 28 hours.

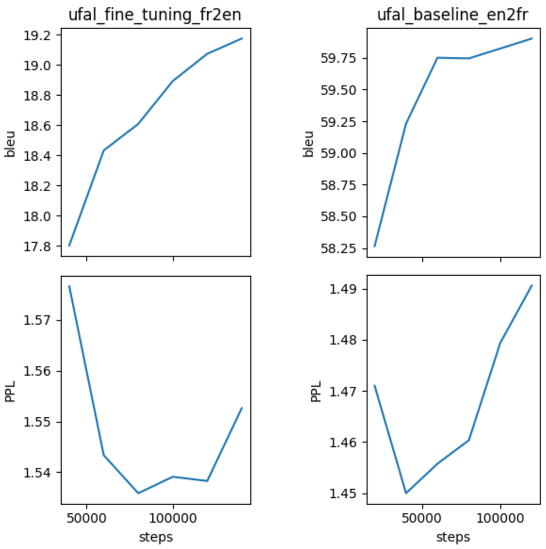


Figure 3: Fine-tuning with UFAL data set FR⇔EN

With the same parameters, the model fine-tuned with UFAL data set had, surprisingly, relatively low scores : we obtained a BLEU score of 18.60 for French→English model and 21.13 as for English→French model (see Figure 3).

4.3 Training and Fine-tuning with Systran Model Studio

SYSTRAN Pure Neural® Server is a multilingual translation platform that offers website translation and localisation features. [5] The server uses Pure Neural® Machine Translation (PNMT®), a commercial engine based on AI and deep learning, launched in 2016. This technology enables neural engines to learn language rules from a given translated text and to produce a translation achieving the current state of the art. An open source neural machine translation system OpenNMT developed by the Harvard NLP group and Systran is available online: http://opennmt.net.

For our work, we used SYSTRAN Pure Neural® Server installed on PAPTAN [6].

We used *characteristic elements* computation (Lebart et al., 1997) implemented in *iTrameur*[7] to compare the results of run2 (generated by SYSTRAN Pure Neural® Server) and run1 (generated by JoeyNMT), using characteristic elements computation (Lebart et al., 1997). In this paper, we discuss the results of FR→EN translation (Table 2). As one can see in Table 2, in the SYSTRAN translation, a sentence always starts with capitalization (“The”, “This”, “In”). Capital letters are also used for acronyms and abbreviations (“BMI”, “VCE”). This can be explained by the default detokenization function of JoeyNMT in detokenizing translation in sub-tokenized form.

The modal verb “must” is overused in the SYSTRAN translation (IndSP = +5) and is never used in the JoeyNMT translation, which tends to prefer the use of the modal verb “should” (Figure 4). The absence of "must" produced by the JoeyNMT system might be due to the large difference of frequencies of both words in training data : 18,462 occurrences of "should" and 4,061 occurrences of "must". The preponderance of "should" in the training corpus has seemingly induced the system to systematically produce the word whenever the system needs to produce a modal verb before a base verb.

We also note that JoeyNMT translation under-

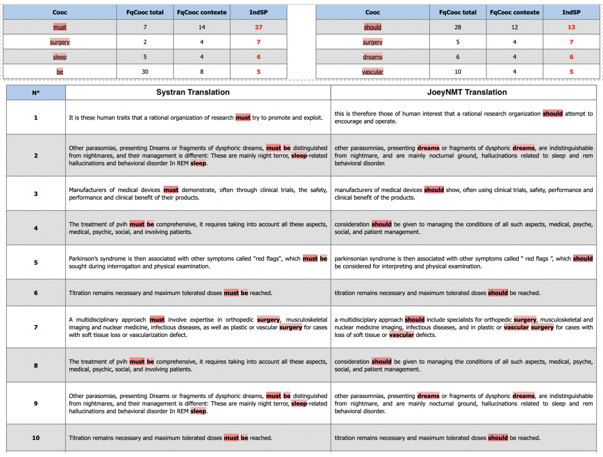
Unit	Fq part	Fq total	IndSP
The	108	108	+33
This	36	36	+12
In	39	39	+12
vaping	28	28	+9
We	23	23	+8
It	19	19	+7
These	18	18	+6
They	16	16	+6
Finally	14	14	+5
must	14	14	+5
Management	9	9	+4
advances	10	10	+4
BMI	11	11	+4
Cancer	10	10	+4
liaison	10	10	+4
However	9	9	+4
VCE	10	10	+4
we	22	71	-4
search	0	10	-4
gc	0	10	-4
bmi	0	13	-5
the	659	1469	-7

Table 2: Characteristic elements of Systran translation (run2) and JoeyNMT translation (run1)

[5] The official product website is available at https://www.systransoft.com/translation-products/systran-pure-neural-server/.

[6] Plateforme pour l’apprentissage profond pour la traduction automatique neuronale, in English: Deep Learning for Machine Translation at Université de Paris-Cité). See the description of the platform on the project website: https://u-paris.fr/plateforme-paptan.

[7] https://itrameur.clillac-arp.univ-paris-diderot.fr



Cooc	FqCooc total	FqCooc contexte	IndSP
must	7	14	27
surgery	2	4	7
sleep	5	4	6
be	30	8	5

Cooc	FqCooc total	FqCooc contexte	IndSP
should	28	12	13
surgery	5	4	7
dreams	6	4	6
vascular	10	4	5

N°	Systran Translation	JoeyNMT Translation
1	It is these human traits that a rational organization of research **must** try to promote and exploit.	this is therefore those of human interest that a rational research organization **should** attempt to encourage and operate.
2	Other parasomias, presenting Dreams or fragments of dysphoric dreams, **must** **be** distinguished from nightmares, and their management is different: These are mainly night terror, **sleep**-related hallucinations and behavioral disorder In REM **sleep**.	other parasomnias, presenting **dreams** or fragments of dysphoric **dreams**, are indistinguishable from nightmare, and are mainly nocturnal ground, hallucinations related to sleep and rem behavioral disorder.
3	Manufacturers of medical devices **must** demonstrate, often through clinical trials, the safety, performance and clinical benefit of their products.	manufacturers of medical devices **should** show, often using clinical trials, safety, performance and clinical benefit of the products.
4	The treatment of pvih **must** **be** comprehensive, it requires taking into account all these aspects, medical, psychic, social, and involving patients.	consideration **should** be given to managing the conditions of all such aspects, medical, psyche, social, and patient management.
5	Parkinson's syndrome is then associated with other symptoms called "red flags", which **must** **be** sought during interrogation and physical examination.	parkinsonian syndrome is then associated with other symptoms called " red flags ", which **should** be considered for interpreting and physical examination.
6	Titration remains necessary and maximum tolerated doses **must** **be** reached.	titration remains necessary and maximum tolerated doses **should** be reached.
7	A multidisciplinary approach **must** involve expertise in orthopedic **surgery**, musculoskeletal imaging and nuclear medicine, infectious diseases, as well as plastic or vascular **surgery** for cases with soft tissue loss or vascularization defect.	a multidisciplary approach **should** include specialists for orthopedic **surgery**, musculoskeletal and nuclear medicine imaging, infectious diseases, and in plastic or **vascular** **surgery** for cases with loss of soft tissue or **vascular** defects.
8	The treatment of pvih **must** **be** comprehensive, it requires taking into account all these aspects, medical, psychic, social, and involving patients.	consideration **should** be given to managing the conditions of all such aspects, medical, psyche, social, and patient management.
9	Other parasomias, presenting Dreams or fragments of dysphoric dreams, **must** **be** distinguished from nightmares, and their management is different: These are mainly night terror, **sleep**-related hallucinations and behavioral disorder In REM **sleep**.	other parasomnias, presenting **dreams** or fragments of dysphoric **dreams**, are indistinguishable from nightmare, and are mainly nocturnal ground, hallucinations related to sleep and rem behavioral disorder.
10	Titration remains necessary and maximum tolerated doses **must** **be** reached.	titration remains necessary and maximum tolerated doses **should** be reached.

Figure 4: Comparison occurrences of "must" and "should" in SYSTRAN and JoeyNMT translations

SYSTRAN translation	JoeyNMT translation
We take stock of knowledge about this addiction and its management.	knowledge about this dependency and the management thereof is a pending state.

Table 3: "we" in SYSTRAN and JoeyNMT translations

uses “we” (IndSP = -4). This finding is interesting because it makes sometimes possible to identify substantial differences between both translations in Table 3.

These results show how training data affects translation results. To our knowledge, SYSTRAN NMT relies upon a broad selection of general texts that do not belong to any single text type, subject field, or register (many of them are translated texts from the web available on https://opus.nlpl.eu). The WMT corpus consists of randomly selected sentences from abstracts and main texts of scientific articles published in medical journals. The articles follow the so-called introduction, methods, results and discussion structure (IMRAD) (Heßler et al., 2020). The selection is not necessarily balanced in terms of represented discourse functions. Thus, we noticed the overuse of “should be” that definitely constrained our translation output (see Figure 4 “should be given”, “should be reached”, “should be considered”, etc.).

5 Discussion

5.1 Degrees of Specialisation

If the Biomedical terminology was indeed present in the testing set (eg "hypertension artérielle pulmonaire","nutriments", "supplémentation en vitamine D"), some sentences were not particularly specialised. For instance, "Le but de cet article est de les résumer de manière relativement exhaustive." is representative of Scientific French for specific purposes but not really of biomedical specialised language. The same holds for the test set from English into French. In view of these observations, it is easy to understand why models trained on more generic data perform so well in this task.

5.2 The performance of gigamodels

We have not submitted translations produced on mBART-50 (Tang et al., 2021), but we compared the translations of our best system (PNS for Pure Neural Server) with those of mBART. [8]. The translation based on mBART produces fluent grammatical sentences but seems to be less specific in the terminology. For instance "vapotage" (*vaping testing*) was translated as *poultry testing* and instead of *vaping frequency* the system produced *pooping frequency*. The terminology is not always consistent or accurate : *hyperthyroïdie frustre* was translated as *rough* (SYSTRAN) or *fruity* (MBART). Oddly enough, with mBART, percentages were literally translated as "per cent" instead of the % symbol.

Figure 5 plots the vocabulary growth curves (VGCs) of the two translated texts. The y axis corresponds to the number of new types and the x axis corresponds to the number of tokens in the translated texts. As can be seen, the two systems have remarkably similar patterns of VGCs, with SYSTRAN PNS slightly above MBART, in spite of the variants we noticed. For the French translation of "keloids", mBART varies between "céloïdes" and "keloïdes", whereas SYSTRAN PNS only produces "chéloïdes".

Measuring specificity indices (Lebart et al., 1997) allowed us to spot differences in the translation. One of the most striking ones was the choice of feminine determiner *la* for *la COVID* in the PNS translations, as evidenced by the specificity of *la COVID* in the two translations (Figure 6). A somewhat belated and debated ruling of the Académie

[8] We used mBART1 through the HuggingFace API (Wolf et al., 2020).https://huggingface.co/facebook/mbart-large-50-many-to-many-mmt

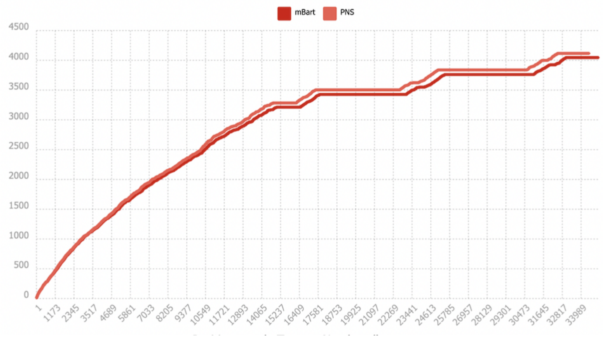


Figure 5: Comparison of Vocabulary Growth Curves in SYSTRAN PNS and mBART translations

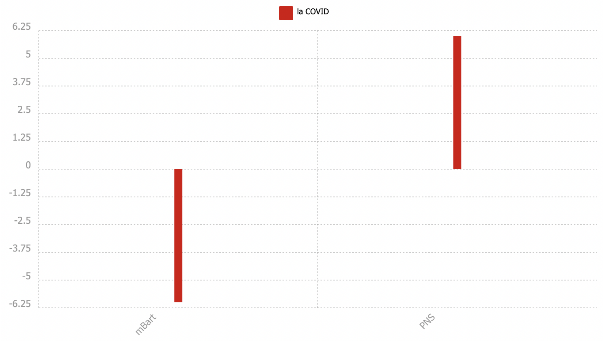


Figure 6: Comparison of Specificity Vocabulary Growth Curves in Systran PNS and mBART translations

française endorsed and imposed "*la*" for the gender of COVID in French. This benign detail probably can be used as a chronological landmark for the training data collection of the two systems: it seems that PNS was trained with more recent French texts. It may also be the case that SYSTRAN has used rule-based normalisation to regularise the output for *la COVID*.

6 Conclusion

This paper presents the SPECTRANS system description for the WMT 2022 biomedical Shared Task. We participated in the English-to-French and French-to-English tasks. We only used the data provided by the organisers but also analysed the translations produced with mBART. We obviously concur with previous research that training data is key. For the MT system, we applied a variety of strategies, toolkit comparison and fine-tuning to compare outcomes of different NMT systems in biomedical translation.

Our contribution mostly lies in the textometric analysis of the output. This allowed us to raise the issue of the role of the variability observed for the gender of COVID in French or for technical terms like "keloids".

Acknowledgements

The SPECTRANS project is funded under the 2020 émergence research project and this publication has emanated from research supported in part by a 2021 research equipment grant from the Scientific Platforms and Equipment Committee (PAPTAN project), both under the ANR grant (ANR-18-IDEX-0001, Financement IdEx Université de Paris).

We gratefully acknowledge support from the CNRS/TGIR HUMA-NUM and IN2P3 Computing Center (Lyon - France) for providing some of the computing and data-processing resources needed for this work.

References

Nicolas Ballier, Dahn Cho, Bilal Faye, Zong-You Ke, Hanna Martikainen, Mojca Pecman, Jean-Baptiste Yunès, Guillaume Wisniewski, Lichao Zhu, and Maria Zimina-Poirot. 2021. The SPECTRANS System Description for the WMT21 Terminology Task. In *EMNLP 2021 SIXTH CONFERENCE ON MACHINE TRANSLATION (WMT21)*, Proceedings of the Sixth Conference on Machine Translation, pages 815–820, Punta Cana, Dominican Republic. ACL.

Nicole Heßler, Miriam Rottmann, and Andreas Ziegler. 2020. Empirical analysis of the text structure of original research articles in medical journals. *PLOS ONE*, 15(10):1–10.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Julia Kreutzer, Jasmijn Bastings, and Stefan Riezler. 2019. Joey NMT: A minimalist NMT toolkit for novices. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 109–114, Hong Kong, China. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Ludovic Lebart, André Salem, and Lisette Berry. 1997. *Exploring textual data*, volume 4. Kluwer Academic.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

SRT's Neural Machine Translation System for WMT22 Biomedical Translation Task

Yoonjung Choi, Jiho Shin, Yonghyun Ryu, Sangha Kim
Samsung Research, Seoul, Republic of Korea
{yj0807.choi, jiho21.shin,
yonghyun.ryu,sangha01.kim}@samsung.com

Abstract

This paper describes the Samsung Research's Translation system (SRT) submitted to the WMT22 biomedical translation task in two language directions: English to Spanish and Spanish to English. To improve the overall quality, we adopt the deep transformer architecture and employ the back-translation strategy for monolingual corpus. One of the issues in the domain translation is to translate domain-specific terminologies well. To address this issue, we apply the soft-constrained terminology translation based on biomedical terminology dictionaries. In this paper, we provide the performance of our system with WMT20 and WMT21 biomedical testsets. Compared to the best model in WMT20 and WMT21, our system shows equal or better performance. According to the official evaluation results in terms of BLEU scores, our systems get the highest scores in both directions.

1 Introduction

Neural Machine Translation (NMT) has shown rapid growth with an encoder-decoder framework, especially Transformer (Vaswani et al., 2017), in recent years. Most of the research focuses on general-purpose translation models since there are a lot of parallel data available. On the other hand, domain-specific translation, which lacks relatively high-quality parallel corpus available, is one of the challenges that need to be solved in the NMT task. To address this issue, there have been several approaches such as finetuning general-purpose models with in-domain data and utilizing in-domain monolingual corpus through back-translation (Yeganova et al., 2021).

In the domain translation, one of the issues is the terminology translation. In the case of domain-specific terms, translation results are often poor because they are relatively infrequent. Yeganova et al. (2021) also mentioned that some domain-specific terms including abbreviations were not translated correctly in previous shared tasks. Moreover, when new terms are introduced such as COVID-19, it is difficult to obtain the correct translation results as they are not in the training data. To handle this issue, we adopt the soft-constrained terminology translation proposed by Molchanov et al. (2021), which provides the terminology constraints of the target language as input to our system with source sentences like a *hint*. These terminology constraints can be obtained from in-domain dictionaries.

In addition, as many domain translation studies, the back-translation strategy (Sennrich et al., 2016) is applied to generate synthetic parallel data from in-domain monolingual corpus. To improve the overall performance of our system, we also employ the Deep Transformer architecture (Bapna et al., 2018) and the ensemble strategy (Sutskever et al., 2014). Moreover, to find better translation results, noisy channel modeling (Yee et al., 2019) and discriminative reranking (Lee et al., 2021) are attempted. Our experiment shows that deep transformer and data augmentation by the back-translation strategy improve the overall performance while the performance is not improved with reranking methods.

The rest of this paper is organized as follows. Section 2 describes the training and test data used in our system, and Section 3 explains our systems including deep transformer and soft-constrained terminology translation. Section 4 describes the details of our training and experimental results of our system; Section 5 presents the official evaluation results. Section 6 is the conclusion of our work.

2 Data

In this section, we present general-domain (out-of-domain) corpus, in-domain corpus, and in-domain terminology dictionaries used as the training data

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 901 - 907
December 7-8, 2022 ©2022 Association for Computational Linguistics

	En2Es	Es2En
General-domain Parallel Corpus	518M	
In-domain Parallel Corpus	3.47M	
In-domain Target-side Monolingual Corpus	2.5M	13.9M
In-domain Dictionaries	132K	
Validataion Data	4,520	
Test Data	921	897

Table 1: Data statistics of the training data, validation data, and test data used in our system.

in our system. For training, all data are tokenized by SentencePiece (Kudo and Richardson, 2018); the vocab size is 32K for each lanaugage. The validation and test data are also described in this section. The statistics of our data are listed in Table 1.

2.1 General-Domain Parallel Corpus

We collect general-domain parallel corpus for English-Spanish from several sources. Some are from WMT News translation task. The data list is as follows: ParaCrawl[1], CommonCrawl[2], Europarl[3], News Commentary[4], and Tatoeba[5].

We also consider two datasets that are provided by organizers: United Nations (UN) Parallel Corpus[6] and UFAL Medical Corpus[7]. The UN Corpus consists of official records and other parliamentary documents of the UN that are in the public domain. In UFAL Medical corpus, it contains not only medical-domain data but also general-domain data; we consider the general-domain data of UFAL as a general-domain parallel corpus in our system.

2.2 In-Domain Parallel Corpus

We use the in-domain data provided by the WMT22 biomedical task organizers.

- Medline Corpus: It contains titles and abstracts of scientific publications. They provide three groups of English-Spanish parallel data: WMT16, WMT19, and WMT22. In WMT16 and WMT19 data, all sentence pairs are already aligned, so we use them without preprocessing process. However, in WMT22 data, all sentences of one abstract are written in one line; thus, after splitting sentences with the sentence splitter provided by Moses[8], only data that matched the number of sentences in both languages are considered as in-domain parallel corpus.

- UFAL Medical Corpus: As we mentioned in Section 2.1, it consists of a general-domain and medical-domain data. The parallel data tagged as the medical-domain are considered in-domain parallel data.

- MeSpEn Corpus: It is the resource for English-Spanish Medical Machine Translation and Terminologies (Villegas et al., 2018). It provides several biomedical and clinical literature data such as IBECS, SciELO, and Pubmed. This corpus contains titles and abstracts from several records. Since all sentences of each abstract are written in one line such as WMT22 Medline corpus, we conduct the same process to extract the parallel corpus.

2.3 In-Domain Monolingual Corpus

In the in-domain parallel corpus, some data are excluded because the number of sentences is not matched between two languages as we menteiond in Section 2.2. In this paper, we use this excluded data as in-domain monolingual data.

Moreover, for the English monolingual corpus, we extract only English data from other language pairs' dataset in Medline corpus and UFAL Medical corpus.

2.4 In-Domain Terminology Dictionary

As we mentioned in Section 1, it is important to translate domain-specific terminologies well in the domain translation. So, we also collect in-domain

[1] https://paracrawl.eu/

[2] https://www.statmt.org/wmt13/training-parallel-commoncrawl.tgz

[3] https://www.statmt.org/wmt13/training-parallel-europarl-v7.tgz

[4] https://www.statmt.org/wmt13/training-parallel-nc-v8.tgz

[5] https://tatoeba.org/en/downloads

[6] https://conferences.unite.un.org/UNCorpus

[7] https://ufal.mff.cuni.cz/ufal_medical_corpus

[8] https://github.com/moses-smt

terminology dictionaries from MeSpEn Glossaries[9] and ClinSpEn-CT[10]. Both are translated by professional medical translators. MeSpEn glossaries contain 125,645 English-Spanish term pairs and CinSpEn-CT sample set includes 7,000 term pairs. We not only utilize in-domain terminology dictionaries as the training data but also use them in the soft-constrained terminology translation.

When the dictionary data is used as the training data, all data in dictionaries is used as it is. However, for the soft-constrained terminology translation, data refinement is required since there are redundant data. It will be described in detail in Section 3.3.

2.5 Vadlidation data

For the validation data, we use the Khresmoi development data. WMT17, WMT18, and WMT19 testset are also used as the validation data.

2.6 Test data

We consider WMT20 and WMT21 "OK" aligned testset as the test data in our system to evaluate the translation quality for the final submission.

3 System Overview

In this section, we describe our system which is based on Transformer architecture (Vaswani et al., 2017). The training details are described in Section 4.1.

3.1 Deep Transformer

Peters et al. (2018) have shown that deeper layers could efficiently extract syntactic and semantic information that could improve the overall performance. Bapna et al. (2018) also have explored deeper encoders for Transformer to improve the translation quality. Several teams that participated in the biomedical shared task last year (Yang et al., 2021; Wang et al., 2021b) have adopted the deep transformer, especially deeper encoders. In this paper, we also adopt the deep transformer architecture which contains 30 encoder layers and 6 decoder layers based on TRANSFORMER-BIG setting (Vaswani et al., 2017).

3.2 Data Augmentation

To augment the in-domain parallel corpus, we adopt back-translation (Sennrich et al., 2016), where the synthetic parallel corpus is generated by translating target-side monolingual data into the source language. Back-translation is one of the effective methods to utilize monolingual data.

In this paper, we first train base models of each direction with the combination of general-domain and in-domain parallel corpus; then, we utilize these trained models to generate source-side sentences from target-side monolingual data.

Moreover, Wang et al. (2021a) present that the overall performance is improved when the in-domain dictionaries are appended to the training corpus. We also consider in-domain terminology dictionaries as the training data.

3.3 Soft-Constrained Terminology Translation

The common approach for the terminology translation is constrained decoding (Hokamp and Liu, 2017), where the translation results are forced to contain pre-specified subsequences, such as the terminology, at decoding time. Since it is the hard-constrained method, it can aggravate the translation quality. Moreover, constrained decoding methods increase the complexity of the decoding process. To address these problems, Dinu et al. (2019) and Molchanov et al. (2021) propose the soft-constrained methods, where pre-specified terminologies are given as input with the source sentence. Although there is no guarantee that translation results always contain these pre-specified terminologies, it can learn a copy behavior at training time without compromising the overall performance.

In this paper, we adopt the soft-constrained strategy of Molchanov et al. (2021) for the terminology translation; that is, we add the desired translation result of the terminology as input with special tokens such as <*term_start*>, <*term_end*>, and <*term_trans*>. Figure 1 presents the example of the revised source sentence including the desired translation result with special tokens. For this, the training corpus should be revised to reflect this input format. First, $N\%$[11] sentence pairs of the training data are randomly extracted and both source and target sentences are tokenized by SpaCy[12] which not only supports tokenization but also provides neural network models for part-of-speech tagging. To obtain the word alignment information between

[9] https://github.com/PlanTL-GOB-ES/MeSpEn_Glossaries

[10] https://zenodo.org/record/6497373#.YxHGtXZBz-j

[11] This is a heuristic value. In this paper, we set it to 15.

[12] https://github.com/explosion/spaCy

Source sentence: Patient had a MI or CVA in last year, or has unstable cardiovascular disease.
Terminology in the source sentence: MI
Desired translation result: IM

New source sentence: Patient had a *<term_start> MI <term_end> IM <term_trans>* or CVA in last year, or has unstable cardiovascular disease.

Figure 1: Example of the revised source sentence for the soft-constrained terminology translation

source and target sentences, the word-aligner[13] is applied. Among aligned words, we only consider *Nouns* as candidates of pre-specified terminologies. In each sentence pair, up to three[14] candidates are randomly selected to provide the desired translation result. Finally, the source sentence is revised by adding a subsequence of the target sentence that is aligned to the selected candidate of the source sentence with special tokens.

For the inference of test data, the biomedical terminology dictionaries described in Section 2.4 are utilized to provide pre-specified terminology information. As we mentioned, terminology dictionaries should be refined. We first remove duplicate terminologies; for instance, if one terminology in the source language is matched with multiple terminologies in the target language, it should be removed since we don't know which of them is the desired translation result. Moreover, if the frequency of the terminology is high in general-domain data, we don't need to consider it. Thus, dictionaries are filtered based on the frequency in general-domain data. For test data, the desired translation results which are from refined dictionaries are added to each source sentence for up to three terminologies, such as the training corpus. If the source sentence in test data doesn't contain any term which is in refined dictionaries, we just input the original source sentence.

3.4 Ensemble

From several NMT studies (Sutskever et al., 2014; Garmash and Monz, 2016; Firat et al., 2016), it has been already shown that ensembling methods can improve the overall performance. In this paper, we conduct the ensemble strategy with the top three models based on our testset for the final submission.

3.5 Reranker

The current NMT system utilizes the beam search approach to generate the final translation result. However, since it is the auto-regressive model, it considers only a limited target context to get the probability of a target token. To address this issue, there are several reranking methods that generate several different hypotheses from the NMT model and rerank them. Since reranking models can consider the entire target context, it can improve the overall performance over the beam search (Lee et al., 2021).

In this paper, we adopt two reranking methods: noisy channel modeling (Yee et al., 2019) and discriminative reranking (Lee et al., 2021). Noisy channel modeling is based on Bayes' rule; it generates translation results based on a backward model and a pre-trained target-side language model. We use a translation model in the opposite direction as a backward model and train transformer language models for the target-side language model. The discriminative reranking model is a transformer architecture that takes the source sentence and the n-best list of output hypotheses as input. It also includes position embeddings and language embeddings for representing two different languages' inputs. As in Lee et al. (2021)'s work, we use XLM-R (Conneau et al., 2020) which is a transformer-based multilingual masked language model as the pre-trained model.

4 Experiments

In this section, we present training details and experimental results of our systems.

4.1 Training details

The baseline models are trained based on TRANSFORMER-BIG setting (Vaswani et al., 2017) which contains 6 encoder layers. We first train baseline models with only general-domain corpus and incrementally train them using in-domain parallel corpus to confirm the effectiveness of in-domain

[13] eflomal, https://github.com/robertostling/eflomal

[14] This is a heuristic value. Based on our training data, we decide this value.

System	Data	En2Es		Es2En	
		WMT20	WMT21	WMT20	WMT21
Best Offiical 20 (Bawden et al., 2020)		0.4672		0.5075	
Best Official 21 (Yeganova et al., 2021)			0.5117		**0.5382**
Baseline	GD	0.4761	0.5134	0.4952	0.5148
	GD+ID	0.4956	0.5305	0.5060	0.5183
Deep Transformer	GD+ID	**0.5174**	0.5485	0.5186	0.5360
+ Data Augmentation	GD+ID+BT+IND	0.5151	0.5523	0.5236	0.5346
+ Ensemble	GD+ID+BT+IND	0.5169	**0.5524**	**0.5255**	0.5332
+ SC Terminology Translation	GD+ID+BT+IND	0.5158	0.5450	0.5216	0.5362
+ Noisy Channel Modeling	GD+ID+BT+IND	0.5143	0.5454	0.5110	0.5255
+ Discriminative Reranking	GD+ID+BT+IND	0.5159	0.5481	-	-

Table 2: BLEU scores on the WMT20 and WMT21 OK aligned test set.

corpus. The deep transformer models which contain 30 encoder layers are trained with the combination of the general-domain and in-domain parallel corpus; based on them, the synthetic data are generated from in-domain monolingual data. Finally, we train the deep transformer models on all corpus: general-domain (GD) and in-domain (IN) parallel corpus, synthetic data (BT), and in-domain dictionary (IND) information. The soft-constrained (SC) terminology translation models are also trained based on deep transformer models with revised training corpus described in Section 3.3. In addition, the ensemble strategy and reranking methods explained in Section 3.5 are applied. For the implementation, we use Fairseq[15], and all models are trained using 8 A100 GPUs. Adam optimizer is used. The batch size is 4K tokens, and the frequency of parameter update is 20. The learning rate, the dropout, and the label smoothing are set to 0.0007, 0.1, and 0.1, respectively. For the inference, the beam size is set to 8. The BLEU scores are calculated using the mt-eval script from Moses (Koehn et al., 2007).

4.2 Experimental results

The experimental results of English to Spanish (En2Es) and Spanish to English (Es2En) directions are shown in Table 2. The baseline models show that the in-domain corpus improves the overall performance in the domain translation. We then apply the deep transformer with the general-domain and in-domain data and it achieves a significant improvement over baseline models. With data augmentation by back-translation of monolingual in-domain data and in-domain dictionaries, there is a slight improvement on average; even though the performance drops slightly in the WMT20 testset of En2Es and WMT21 testset of Es2En, it improves more in other testset of each direction.

En2Es	WMT20	WMT21
Plain Testset	0.5158	0.5450
Revised Testset	**0.5325**	**0.5505**
Es2En	**WMT20**	**WMT21**
Plain Testset	0.5216	0.5362
Revised Testset	**0.5294**	**0.5472**

Table 3: BLEU scores of soft-constrained terminology translation models on plain testsets and revised testsets with soft-constrained terminologies.

The ensemble models show better performance than a single model in general.

In the soft-constrained terminology translation, the performance is slightly improved in one testset while the performance is decreased in the other testset in each direction. Since the soft-constrained terminology translation models are trained with revised corpus, the testset also should be revised by adding desired translation results with special tokens in order to evaluate the performance accurately. Table 3 shows BLEU scores of soft-constrained terminology translation models on plain testsets and revised testsets which contain desired translation results. We observe that soft-constrained terminology translation models are more effective when the desired translation results of some terminologies are given such as training corpus.

As we mentioned in Section 3.5, two reranking methods are adopted, but as a result, the overall

[15] https://github.com/facebookresearch/fairseq

System	En2Es	Es2En
Best Official	0.5235	0.6045
SRT run1	0.5214	0.5954
SRT run2	0.5196	0.5943
SRT run3	**0.5235**	**0.6045**

Table 4: Official BLEU scores of our submissions for WMT22 biomedical task.

performance is not improved. (The discriminative reranking is experimented only on En2Es.)

Since there is no improvement with two reranking methods, we exclude their results in our final submissions. Our final submissions are results of data augmentation, ensembling models, and soft-constrained terminology translation.

5 Official Evaluation Results

The official evaluation results of our submissions (SRT) for WMT 2022 biomedical translation task are shown in Table 4. All our submissions show the best BLEU scores.

6 Conclusion

This paper presents the Samsung Research's Translation system (SRT) for the WMT22 biomedical translation shared task in two language directions: English to Spanish and Spanish to English. We perform experiments with several strategies such as deep transformer, data augmentation, soft-constrained terminology translation, ensembling models, and reranking methods. Our experiments show the effectiveness of each strategy. The deep transformer, data augmentation, and ensemble strategies improve effectively the overall performance in the domain translation. Moreover, we present that the soft-constrained terminology translation is a reasonable method to achieve good performance in the domain translation. Our systems show the best BLEU scores in the official evaluation results.

References

Ankur Bapna, Mia Chen, Orhan Firat, Yuan Cao, and Yonghui Wu. 2018. Training deeper neural machine translation models with transparent attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3028–3033, Brussels, Belgium. Association for Computational Linguistics.

Rachel Bawden, Giorgio Maria Di Nunzio, Cristian Grozea, Inigo Jauregi Unanue, Antonio Jimeno Yepes, Nancy Mah, David Martinez, Aurélie Névéol, Mariana Neves, Maite Oronoz, Olatz Perez-de Viñaspre, Massimo Piccardi, Roland Roller, Amy Siu, Philippe Thomas, Federica Vezzani, Maika Vicente Navarro, Dina Wiemann, and Lana Yeganova. 2020. Findings of the WMT 2020 biomedical translation shared task: Basque, Italian and Russian as new additional languages. In *Proceedings of the Fifth Conference on Machine Translation*, pages 660–687, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Orhan Firat, Baskaran Sankaran, Yaser Al-onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas. Association for Computational Linguistics.

Ekaterina Garmash and Christof Monz. 2016. Ensemble learning for multi-source neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1409–1418, Osaka, Japan. The COLING 2016 Organizing Committee.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Ann Lee, Michael Auli, and Marc'Aurelio Ranzato. 2021. Discriminative reranking for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7250–7264, Online. Association for Computational Linguistics.

Alexander Molchanov, Vladislav Kovalenko, and Fedor Bykov. 2021. PROMT systems for WMT21 terminology translation task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 835–841, Online. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Marta Villegas, Ander Intxaurrondo, Aitor Gonzalez-Agirre, Montserrat Marimon, and Martin Krallinger. 2018. The mespen resource for english-spanish medical machine translation and terminologies: Census of parallel corpora, glossaries and term translations. In *LREC MultilingualBIO: Multilingual Biomedical Text Processing. ELRA*.

Weixuan Wang, Wei Peng, Xupeng Meng, and Qun Liu. 2021a. Huawei AARC's submissions to the WMT21 biomedical translation task: Domain adaption from a practical perspective. In *Proceedings of the Sixth Conference on Machine Translation*, pages 868–873, Online. Association for Computational Linguistics.

Xing Wang, Zhaopeng Tu, and Shuming Shi. 2021b. Tencent ai lab machine translation systems for the wmt21 biomedical translation task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 874–878, Online. Association for Computational Linguistics.

Hao Yang, Zhanglin Wu, Zhengzhe Yu, Xiaoyu Chen, Daimeng Wei, Zongyao Li, Hengchao Shang, Minghan Wang, Jiaxin Guo, Lizhi Lei, chuanfei xu, Min Zhang, and Ying Qin. 2021. Hw-tsc's submissions to the wmt21 biomedical translation task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 879–884, Online. Association for Computational Linguistics.

Kyra Yee, Yann N. Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5695–5700. Association for Computational Linguistics.

Lana Yeganova, Dina Wiemann, Mariana Neves, Federica Vezzani, Amy Siu, Inigo Jauregi Unanue, Maite Oronoz, Nancy Mah, Aurélie Névéol, David Martinez, Rachel Bawden, Giorgio Maria Di Nunzio, Roland Roller, Philippe Thomas, Cristian Grozea, Olatz Perez-de Viñaspre, Maika Vicente Navarro, and Antonio Jimeno Yepes. 2021. Findings of the WMT 2021 biomedical translation shared task: Summaries of animal experiments as new test set. In *Proceedings of the Sixth Conference on Machine Translation*, pages 664–683, Online. Association for Computational Linguistics.

Examining Large Pre-Trained Language Models for Machine Translation: What You Don't Know About It

Lifeng Han[1], **Gleb Erofeev**[2], **Irina Sorokina**[2], **Serge Gladkoff**[2], and **Goran Nenadic**[1]
[1] The University of Manchester, UK
[2] Logrus Global, Translation & Localization
lifeng.han, g.nenadic@manchester.ac.uk
gleberof, irina.sorokina, serge.gladkoff@logrusglobal.com

Abstract

Pre-trained language models (PLMs) often take advantage of the monolingual and multilingual dataset that is freely available online to acquire general or mixed domain knowledge before deployment into specific tasks. Extra-large PLMs (xLPLMs) are proposed very recently to claim supreme performances over smaller-sized PLMs such as in machine translation (MT) tasks. These xLPLMs include Meta-AI's wmt21-dense-24-wide-en-X (2021) and NLLB (2022). *In this work, we examine if xLPLMs are absolutely superior to smaller-sized PLMs in fine-tuning toward domain-specific MTs.* We use two different in-domain data of different sizes: commercial automotive in-house data and **clinical** shared task data from the ClinSpEn2022 challenge at WMT2022. We choose popular Marian Helsinki as smaller sized PLM and two massive-sized Mega-Transformers from Meta-AI as xLPLMs.

Our experimental investigation shows that 1) on smaller sized in-domain commercial automotive data, xLPLM wmt21-dense-24-wide-en-X indeed shows much better evaluation scores using SACREBLEU and hLEPOR metrics than smaller-sized Marian, even though its score increase rate is lower than Marian after fine-tuning; 2) on relatively larger-size well prepared clinical data fine-tuning, the xLPLM NLLB **tends to lose** its advantage over smaller-sized Marian on two sub-tasks (clinical terms and ontology concepts) using *ClinSpEn offered metrics* METEOR, COMET, and ROUGE-L, and totally lost to Marian on Task-1 (clinical cases) on *all official metrics* including SACREBLEU and BLEU; 3) **metrics do not always agree** with each other on the same tasks using the same model outputs; 4) clinic-Marian ranked No.2 on Task-1 (via SACREBLEU/BLEU) and Task-3 (via METEOR and ROUGE) among all submissions.

1 Introduction

Owing to the recent development of neural machine translations (NMTs) (Kalchbrenner and Blunsom, 2013; Cho et al., 2014; Bahdanau et al., 2014; Akhbardeh et al., 2021; Han, 2022a), especially the self-attention based Transformer learning structures (Devlin et al., 2019; Vaswani et al., 2017), pre-trained language models (PLMs) have been dominant in natural language understanding (NLU) and natural language processing (NLP) tasks. These applications include Long-Short Term Memory (LSTM) and BERT (Pre-training of Deep Bidirectional Transformers) based models to text mining (Dernoncourt et al., 2017; Wu et al., 2022), question-answering (Dong et al., 2021), reading comprehension (Schlegel, 2021), and summarisation (Perez-Beltrachini and Lapata, 2021), etc., in addition to MT (Han et al., 2021a; Han, 2022b; Han and Gladkoff, 2022).

PLMs often have a large amount of trainable parameters for downstream applications. For instance, in translation task, the popular Marian NMT (Junczys-Dowmunt et al., 2018) pre-trained by Microsoft Translator team [1] on OPUS [2] (Tiedemann, 2012) multilingual corpus has 7.6 million parameters, which can still be fine-tuned on Google's Colab or AWS at virtually no cost. However, very recent work has shown much larger PLMs that have much more parameters than smaller models, e.g. the multi-lingual Transformer model submitted to WMT2021 shared task by Meta-AI research group "wmt21-dense-24-wide-en-x"(WMT21fb) (Tran et al., 2021), which has 4.7 billion parameters, i.e. 618 times bigger than Marian, and does not fit into regular GPUs. In this year, Meta-AI published another model NLLB (NLLB Team et al., 2022) that has 54.5 billion parameters and covers 200 languages in the full model. From now on, we name

[1] https://translator.microsoft.com
[2] https://opus.nlpl.eu

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 908 - 919
December 7-8, 2022 ©2022 Association for Computational Linguistics

both “wmt21-dense-24-wide-en-x” and NLLB as Meta-AI’s *Mega-Transformer* models. Meta-AI’s Mega-Transformer (WMT21fb) has claimed the best performing system on 10 out of 14 language pairs in WMT2021 shared task including winning bilingual-trained models.

In this work, we raise the question whether extra-large PLMs (xLPLMs) such as Meta-AI’s Mega-Transformers have absolute superiority in NMT tasks on domain-specific fine-tuning. We prepare experimental investigation on two different data set to answer this question. One is our specific automotive domain in-house commercial data and the other is clinical domain data from ClinSpEn2022 challenge task we attended which is affiliated with WMT2022 [3].

We set up the following *hypothesis* and *research questions*. Our ***hypothesis*** is: xLPLMs do not absolutely demonstrate superiority over smaller sized PLMs in NMT fine-tuning and it shall depends on specific tasks deployed including domain topic, size of available in-domain data, and performance-cost trad-off.

From this hypothesis we derive two *research questions (**RQs**)*: 1) Do xLPLMs always demonstrate better performances in NMT over smaller sized PLMs for domain fine-tuning? 2) if not, in what situations?

To the best of our knowledge, this is the first published work that has been carried out in the field on fine-tuning Meta-AI’s extra-large multilingual PLM Maga-Transformers, and in translating specialised automotive and clinical data.

The rest of the paper is organised as below: Section 2 introduces more details on related work to ours including PLMs and fine-tuning in automotive and clinical domains, Section 3 describes our initial model settings including deployed baseline models, Section 4 presents our experimental evaluation carried out on our in-house commercial automotive domain data, Section 5 describes our system submission to ClinSpEn Biomedical-MT challenge task at WMT2022 on clinical data, and Section 6 gives our conclusion and future work plan.

2 Related Work

Fine-tuning PLMs has been in practice towards different domain applications in recent years. For instance, Wang et al. (2021) carried out experimental investigation on fine-tuning PLMs for conversational recommendation system, Chakraborty et al. (2020); Gu et al. (2021); Lee et al. (2019); Alsentzer et al. (2019) built biomedical and clinical domain pre-trained models using BERT structure and PubMed data on scientific publications, and then Wu et al. (2022); Han et al. (2022a) developed new machine learning structures using PLM Transformer and BERT as encoders in concatenation with statistical graph-based conditional random fields (CRFs) as decoders for clinical text mining.

However, aforementioned work did not deploy extra-large PLMs in a scale as Meta-AI’s multilingual Mega-Transformers. For example, the PLMs (Transformer-CRFs) deployed by Wu et al. (2022) as baseline have around 42 million of trainable parameters, which set is already relatively large, even though it is still far from Mega-Transformers’ 4.7 billion and 54.5 billion parameters.

Regarding PLM applications in automotive domain, the only recent work we found is from Romell and Curman (2022), who tested the DistilBERT and XLM-RoBERTa PLMs for text classification task using Swedish truck manufacturer data, instead of MT.

There are also researchers working on the overview of model comparability, bench-marking, and fine-tuning methodologies regarding larger scale PLMs, e.g., from Aßenmacher (2021); Ruder (2021).

Overall, none of the work mentioned before has investigated into extra-large Mega-Transformer level PLMs (xLPLMs) for NMT in automotive and clinical domains, especially their comparisons to smaller sized PLMs.

3 Initial Model Settings

To investigate into PLMs with fine-tuning for specialised domain NMT from different scales, we firstly deploy two of such models in different sizes from a multilingual setting. The first one is the popular Marian NMT model developed in C++ since 2018 using deep RNN and Transformer (Junczys-Dowmunt et al., 2018). It is mostly maintained by the Microsoft Translator team and features with efficiency, fast training, and state-of-the-art NMT architectures [4]. This PLM has a smaller sized 7.6 million trainable parameters.

[3] The 7th Conference on MT https://www.statmt.org/wmt22/

[4] available at https://marian-nmt.github.io

The second one we use for fine-tuning is one of the extra-large PLMs (*xLPLMs*) Meta-AI's Mega-Transformer "wmt21.dense-24-wide.En-X" (Tran et al., 2021) developed for WMT2021 shared task on multilingual MT, which was submitted to 14 language pairs and claimed the best on 10 of them [5]. It has 4.7 billion trainable parameters, which is super large in comparison to Marian model. In the later section (5), we will explain another Mega-Transformer Model NLLB developed in this year by Meta-AI and deploy it for our ClinSpEn2022 shared task submission on clinical domain.

4 Model Fine-Tuning and Comparison on Commercial Automotive Data

4.1 In-house Corpus and Hardware

At the development stage, we use our in-house prepared domain-specific commercial corpus from automotive field. We split our data set into 90% vs 10% for fine-tuning and testing respectively and make sure that the test data is not seen during the fine-tuning / development stage [6]. We use a larger GPU from NVIDIA A100 with 80GB VRAM for our experiments because of the much higher computational powers the Mega-Transformer model requires.

4.2 Our Evaluation Setup

BLEU (Papineni et al., 2002) has always been criticised by researchers on its reliability. This includes very recent work by Freitag et al. (2021), which demonstrates that BLEU has closer correlation to lower quality crowd sourced human evaluation then to expert based human evaluation, and by Han et al. (2021a), which investigation on Chinese-English NMT shows that BLEU score fails to reflect the real quality differences between NMT systems especially on translating multi-word expressions (MWEs) and terms (Han et al., 2020).

Furthermore, BLEU scores can be very different caused by configurations, such as tokenisation and normalisation strategies applied to the reference text which can lead to 1.8 margin of difference reported by (Post, 2018). In light of these findings, we adopt two alternative evaluation metrics, i.e. SACREBLEU (Post, 2018) and hLEPOR (Han et al., 2013b; Erofeev et al., 2021; Han et al., 2021b) that we will give further details about.

4.2.1 Revisiting SACREBLEU

SACREBLEU is developed by the work from Post (2018) and is maintained online in its Python version [7]. The author discussed the uncertainty regarding reporting BLEU scores by MT researchers. This is involved in many parameter settings when using BLEU metric including number of references, length penalty computation on multi-references, maximum n-gram, and smoothing applied to 0-count n-grams. Because of such variations, when MT researchers report the BLEU scores from their system, "the BLEU" score actually cannot be reproduced in many cases due to lack of detailed technical description of encoder, etc. .

To address these issues, SACREBLEU added some constrains while using BLEU metric. These include the applying of its own metric-internal pre-processing for detokenised system outputs, the avoiding of user handling reference set via automatically downloading from WMT, and the export of a summary on settings used.

4.2.2 Revisiting hLEPOR

hLEPOR is an augmented metric for automatic MT evaluation which was firstly proposed in WMT2013 Metrics shared task (Han et al., 2013a,b) and was reported as one of the best performing metrics at both system level (Macháček and Bojar, 2013) and segment level (Graham et al., 2015) [8]. It is calculated via a weighted harmonic mean of several main factors including sentence length penalty, position difference penalty, precision, and recall. Furthermore, there are more weighting parameters among all the sub-factors. Let's see the brief formulas below:

$$
h\text{LEPOR} = Harmonic(w_{LP}LP, \\ w_{NPosPenal}NPosPenal, w_{HPR}HPR)
$$

where *LP* is the sentence length penalty factor and is calculated as:

[5] package "wmt21.dense-24-wide.En-X" available at https://github.com/facebookresearch/fairseq/tree/main/examples/wmt21

[6] Because this is a commercial corpus, we do not give much details on it but this does not affect the experimental findings we achieved

[7] available at https://github.com/mjpost/sacrebleu

[8] The python version is available at https://pypi.org/project/hLepor/ and the original Perl code at https://github.com/poethan/LEPOR

$$\text{LP} = \begin{cases} e^{1-\frac{Length_{ref}}{Length_{hyp}}} \; if \; Length_{hyp} < Length_{ref} \\ 1 \; if \; Length_{hyp} = Length_{ref} \\ e^{1-\frac{Length_{hyp}}{Length_{ref}}} \; if \; Length_{hyp} > Length_{ref} \end{cases}$$

Then, n-gram based position difference penalty (NPD) is used to measure the word position and order difference among matched words between system output and reference translation ($MatchN_{hyp}$ and $MatchN_{ref}$).

$$NPosPenal = e^{-NPD}$$

$$NPD = \frac{1}{Length_{hyp}} \sum_{i=1}^{Length_{hyp}} |PD_i|$$

$$|PD_i| = |MatchN_{hyp} - MatchN_{ref}|$$

Finally, the weighted harmonic mean of precision and recall is calculated using this formula.

$$HPR = \frac{(\alpha + \beta) Precision x Recall}{\alpha Precision + \beta Recall}$$

$$Precision = \frac{Aligned_{num}}{Length_{hypothesis}}$$

$$Recall = \frac{Aligned_{num}}{Length_{reference}}$$

hLEPOR is an extended version of the original LEPOR metric (Han et al., 2012; Han, 2014). hLEPOR also has a latest customised version named cushLEPOR which uses automatic hyper-parameter optimisation framework Optuna (Akiba et al., 2019) to achieve better and easier feature weights fine-tuning towards specific language pairs and domains in practice. It was reported as one of the best performing metrics in WMT2021 (Erofeev et al., 2021; Han et al., 2021b) on the officially-ranked language pairs English-German and Chinese-English on News domain, and English-Russian on TED talk data (Freitag et al., 2021) where human expert level evaluations were available. hLEPOR is also gaining popularity in other NLP task evaluations, e.g. language generation (NLG) (Novikova et al., 2017; Gehrmann et al., 2021; Marzouk, 2021), language understanding (NLU) (Ruder et al., 2021), text summarization (ATS) (Bhandari et al., 2020), and searching (Liu et al., 2021).

4.3 Evaluation Results

The evaluation scores using SACREBLEU and hLEPOR are shown in Table 1 and 2 respectively. From Table 1, we can see that the fine-tuning has successfully improved each single n-gram precision score in SACREBLEU for both Marian and Mega-Transformer models, leading to an overall 150.14% and 75.81% score increasing. Similarly, Table 2 shows that our in-domain fine-tuning improved hLEPOR scores on Marian and Mega-Transformer models via 32.16% and 26.01%.

Like BLEU, SACREBLEU is precision based metric. The very large margin evaluation score increases in SACREBLEU (150.14% and 75.81%) indicates that according to reference translation, our fine-tuned models produce more fluent output than the baseline in this domain specific test set. Unlike SACREBLEU, hLEPOR is an augmented metric with comprehensive factors, including recall and positional difference penalty, in addition to precision. The large margins of hLEPOR score increase, i.e. 32.16% and 26.01% tell that the fine-tuned models can also have more adequate translation towards this domain, in addition to maintaining higher fluency.

In summary, the fine-tuning of these two PLMs has demonstrated evaluation score improvement with large margins in commercial domain data. xLPLM Mega-Transformer has much higher SACREBLEU evaluation score than Marian before fine-tuning, 39.12 vs 19.64, which indicates its larger amount of knowledge acquired. However, after fine-tuning, the SACREBLEU scores of them are much closer, 50.33 vs 45.20. This means that fine-tuning of smaller sized PLM for this commercial data is far more effective than the xLPLM Mega-Transformer from computation and time cost point of view, as well as the cost of computational power itself, since supercomputer time is much more expensive.

This partially verifies our assumption that xLPLMs do not always win smaller sized PLMs in practical applications when computational cost is in place and when time is constrained.

To further investigate our research questions, we carry out another experimental evaluation on clinical domain data via attending the ClinSpEn2022 shared task challenge which will be detailed in the next section.

	Marian					
	uni-gram	bi-gram	tri-gram	4-gram	BP	Overall
Before fine-tuning	19.64	10.96	4.56	2.00	1.0	7.38
After fine-tuning	45.20	24.54	14.44	8.69	0.96	18.46 (↑150.14%)
	Mega-Transformer (wmt21fb)					
	uni-gram	bi-gram	tri-gram	4-gram	BP	Overall
Before fine-tuning	39.12	18.81	9.78	5.23	1.0	13.93
After fine-tuning	50.33	30.14	19.47	12.85	0.99	24.49 (↑75.81%)

Table 1: SACREBLEU score comparisons on the MT test set: before vs after fine-tuning

	Marian	Mega-Transformer
Before fine-tuning	36.91	47.55
After fine-tuning	48.78	59.92
Rate(↑)	32.16%	26.01%

Table 2: hLEPOR score comparisons on the MT test set: before vs after fine-tuning

5 Submission to ClinSpEn at WMT22

In this section, we introduce our system submissions to Biomedical-MT task in WMT2022. In this task, we attended the affiliated clinical domain machine translation on Spanish-English language pair (ClinSpEn) task [9], which is hosted in CodaLab (Pavao et al., 2022) [10].

The aim of this task is to promote the development of MT models on medical domain via three sub-tasks: 1) Clinical Cases (CC): on 202 COVID-19 clinical case reports; 2) Clinical Terms (CT): using more than 19K parallel terms extracted from biomedical literature and electric health records (EHRs); 3) Ontology Concepts (OC): using more than 2K parallel concepts from biomedical ontology. The translation direction on these three sub-tasks are EN→ES, EN←ES, and EN→ES respectively.

5.1 Corpus Used

In addition to the official corpora prepared by the ClinSpEn organisers, we used some external corpora for our model fine-tuning. This is because that neural-network based machine learning models are data dependent while the officially offered parallel sample sentences are very limited. We found useful biomedical Spanish-English corpora described in (Névéol et al., 2018) from WMT[11], and MeSpEn corpora from (Villegas et al., 2018)[12], which include Spanish Bibliographical Index in Health Sciences (IBECS), Scientific Electronic Library Online (SciELO), and U.S. National Library of Medicine (PubMed and MedlinePlus). However, due to the time restriction for this shared task, we only managed to get 250,000 aligned pairs from IBECS after careful preparation, which is a bibliographical data collecting scientific articles from different fields of health sciences, maintained by the Spanish National Health Sciences Library.

5.2 Adaptations on xLPLM: NLLB

Two systems we submitted to ClinSpEn2022 are clinic-Marian and clinic-NLLB (NLLB Team et al., 2022). We reported our clinic-WMT21fb model outputs in a followup work (Han et al., 2022b) (also due to the time restriction). Some training parameters and training logs for clinic-Marian are listed below:

- batch size = 64
- gradient accumulation steps = 1
- weight decay = 0.01
- learning rate = 2e-5
- number of training epochs = 1
- number of examples = 225,000

NLLB (No Language Left Behind) is another extra-large PLM model built by Meta-AI freshly in

[9] https://temu.bsc.es/clinspen/

[10] https://codalab.lisn.upsaclay.fr/competitions/6696

[11] https://github.com/biomedical-translation-corpora

[12] https://zenodo.org/record/3562536

this year [13], which was targeting low-resource languages via knowledge transfer from high-resource ones, and Spanish is among the high-resource languages covered by NLLB [14]. NLLB-200 has a total of 54.5 billion parameters in its full model as the authors mentioned. In this shared task, we applied the distilled version of NLLB, i.e. the "NLLB-200-distilled-1.3B" which still has 1.3 billion trainable parameters [15]. As Meta-AI's "wmt21.dense-24-wide.en-X" model we used in the earlier section, we call NLLB-distilled as one of their *Mega-Transformers*.

Some fine-tuning parameters for NLLB-distilled are listed below:

- batch size = 24
- gradient accumulation steps = 8
- weight decay = 0.01
- learning rate = 2e-5
- number of training epochs = 1
- encoder-decoder layers = 24+24

The fine-tuned clinic-NLLB model has relatively apparent evaluation score increase using SacreBLEU in comparison to baseline model on both translation directions, as shown in Table 3, for EN→ES and ES→EN in the upper and middle parts of the table with increasing rate 11.74% and 9.70% respectively. This demonstrates that that fine-tuning was successful.

Interestingly, if we fine-tune the model in one direction and carry out the inference translation in the opposite direction, the model performance will have a big drop even though it is the same language pair. This tells that pre-trained LMs lose their generalisation after fine-tuning. For instance, in the bottom of Table 3, we demonstrate that if the model is fine-tuned in English-to-Spanish direction and the inference test is carried out in Spanish-to-English direction, the overall SacreBLEU score has a 14.37% drop in comparison to without fine-tuning. So, we carried out fine-tuning on both translation directions for the system submission to three sub-tasks at ClinSpEn2022.

5.3 Official Evaluation Metrics

The official evaluation metrics used by ClinSpEn2022 shared task are METEOR (Banerjee and Lavie, 2005), SacreBLEU (Post, 2018), COMET (Rei et al., 2020), BLEU-HF (HuggingFace) (Papineni et al., 2002), and ROUGE-L-F1 (Lin, 2004). Among these, METEOR is a metric using both precision and recall not only on word surface level but also introducing paraphrasing features. COMET was proposed recently by taking advantage of cross-lingual PLMs using knowledge from both source and target languages. ROUGE was originally designed for text summarisation evaluation using n-gram co-occurrences, while ROUGE-L added the Longest Common Sub-sequence (LCS) feature from translation study.

5.4 Evaluation Scores on Three Tasks

We present the MT evaluation scores using five official metrics through CodaLab platform on the three sub-tasks in Table 4, for translating clinical cases, clinical terms, and ontology concepts. The two fine-tuned models are clinic-Marian and clinic-NLLB (one of the Mega-Transformers). From this shared task evaluation outcomes, the xLPLM clinic-NLLB starts to lose its comparisons to far smaller-sized clinic-Marian in Task-2 (CT) and 3 (OC), especially on METEOR and ROUGE-L scores but also on COMET (OC). What is very noticing is that clinic-Marian has an overall win on Task-1 (CC) via all evaluation metrics.

From the evaluation results on Task 2 and 3, i.e. CT and OC, we can see that the evaluation metrics do not agree with each other always. For instance, clinic-Marian wins METEOR and ROUGE-L on Task 2 but loses on other metrics, while clinic-NLLB wins SacreBLEU and BLEU-HF on Task 3 but loses on other metrics. This phenomenon is very interesting which tells that variation metrics from BLEU including BLEU-HF and SacreBLEU tend to not agree with other metric families including METEOR, COMET, and ROUGE-L. Furthermore, the same metric does not always agree with itself on different tasks, or the two MT models perform differently across tasks. For instance, COMET score says clinic-Marian and clinic-NLLB wins Task 3 (0.9495) and 2 (1.0290) respectively. Due to the time restriction from this shared task and the limited computational resource we have, our second model (clinic-NLLB) was submitted after the official deadline.

[13] The project page https://ai.facebook.com/research/no-language-left-behind/

[14] Models available at https://huggingface.co/docs/transformers/model_doc/nllb

[15] https://huggingface.co/facebook/nllb-200-distilled-1.3B

	English-to-Spanish (tune+test)					
	uni-gram	bi-gram	tri-gram	4-gram	BP	Overall
Before fine-tuning	65.93	45.51	33.71	25.44	1.0	40.05
After fine-tuning	70.25	50.58	38.78	30.17	0.99	44.75 (↑11.74%)
	Spanish-to-English (tune+test)					
	uni-gram	bi-gram	tri-gram	4-gram	BP	Overall
Before fine-tuning	65.36	42.54	30.58	22.60	1	37.23
After fine-tuning	68.51	46.27	34.07	25.76	1	40.84 (↑9.70%)
	English-to-Spanish (tune) & Spanish-to-English (test)					
	uni-gram	bi-gram	tri-gram	4-gram	BP	Overall
Before fine-tuning (es2en)	65.36	42.54	30.58	22.60	1	37.23
After *Reverse* fine-tuning	58.17	36.48	25.85	18.84	1.0	31.88 (↓14.37%)

Table 3: SACREBLEU score comparisons using NLLB: baseline vs fine-tuned in clinical domain.

This experimental investigation shows that with carefully prepared and larger amount of domain specific data for fine-tuning, the xLPLMs tend to lose its advantage over smaller sized PLMs using several automatic metrics. Thus it further verifies our hypothesis and research questions.

5.5 Comparisons to Other Teams

In the officially valid submissions (before the shared task deadline ended) for three tasks, there are four teams for Task-1 and Task-3 including Avellana Translation, DtranX, Optum and ours [16]. In addition to these four teams, Task-2 has another team Huawei, making it in-total five teams. Optum and Huawei have both multiple submissions/runs while other teams submitted one run. Our submission clinic-Marian ranked number 2 in both Task-1 and Task-3 via SACREBLEU/BLEU and METEOR/ROUGE respectively, as in Table 5 underlined. There are four runs from Optum team for both Task-1/3 and single submission by other teams. Table 5 includes the best submission from Optum. There is a little difference in the last digit of the evaluation scores between our own record (Table 4) and the official record (Table 5), which is because that we rounded the last digit scores while the official ones did not. This result shows that metrics tend to not agree with each others in many cases. For instance, on Task-3, our clinic-Marian has very similar score to DtranX on METEOR (0.6261 vs 0.6275) only from the third digit which is a metric using paraphrase and semantic similarity features; however, the score difference on BLEU is so large (39.10 vs 58.24) via SACREBLEU which rises the issue again on the credibility of BLEU metric. There are not many teams submitting their results into this clinical domain machine translation task in comparison to the traditional news domain MT task, which indicates that it is still a relatively new domain and calls for more attentions from MT researchers in the future.

6 Discussion and Future Work

In this work, we carried out experimental investigations on if extra-large pre-trained language models (PLMs) always demonstrate superiority over much smaller-sized PLMs using two domain specific data. The first experimental results using Marian vs "wmt21.dense-24-wide.En-X" shows that even though xLPLM still perform better evaluation scores in comparison to much smaller sized Marian, their score difference is much smaller after fine-tuning and the xLPLM costs more than smaller PLM from performance-cost trade-off point of view in practical applications, e.g. for language service providers (LSPs). The second experimental results using clinical data show that with carefully prepared certain amount of fine-tuning data (250k sentence pairs), the xLPLM NLLB even loses with its evaluation score in comparison to smaller PLM Marian in Task 1 "clinical cases" over all automatic metrics used, and in Task 2 "clinical terms" and 3 "ontology concepts" on partial of the automatic evaluation metrics officially used by ClinSpEn2022. Finally, our system submission clinic-Marian ranked the second place using SACREBLEU/BLEU for

[16] https://statmt.org/wmt22/biomedical_results.pdf

	clinic-Marian				
MT	SACREBLEU	METEOR	COMET	BLEU-HF	ROUGE-L-F1
Task-I: clinical cases	*38.18*	*0.6338*	*0.4237*	*0.3650*	*0.6271*
Task-II: clinical terms	26.87	*0.5885*	0.9791	0.2667	*0.6720*
Task-III:clinical concepts	39.10	*0.6262*	*0.9495*	0.3675	*0.7688*
	clinic-NLLB (Mega-Transformers)				
MT	SACREBLEU	METEOR	COMET	BLEU-HF	ROUGE-L-F1
Task-I: clinical cases	37.74	0.6273	0.4081	0.3601	0.6193
Task-II: clinical terms	*28.57*	0.5873	*1.0290*	*0.2844*	0.6710
Task-III: ontology concepts	*41.63*	0.6072	0.9180	*0.3932*	0.7477

Table 4: Evaluation Scores using Official CodaLab Platform from ClinSpEn2022 Benchmark on Fine-tuned Models. *italic* scores indicate winner on the specific task using the specific metric (last digit rounded).

	Task-1: Translating Clinical Cases				
Teams	SACREBLEU	METEOR	COMET	BLEU	ROUGE
DtranX	*41.06*	0.6633	*0.4610*	*0.3926*	*0.6490*
Logrus-UoM (ours)	38.17	0.6337	0.4237	0.3650	0.6270
Optum(run4)	38.12	0.6447	0.4425	0.3642	0.6285
Avellana Translation	36.64	*0.6637*	0.3920	0.3519	0.6333
	Task-3: Translating Ontology Concepts				
Teams	SACREBLEU	METEOR	COMET	BLEU	ROUGE
DtranX	*58.24*	*0.6275*	*1.2496*	0.5724	*0.7839*
Optum(run4)	44.97	0.5880	1.1197	0.4396	0.7479
Logrus-UoM (ours)	39.10	0.6261	0.9494	0.3674	0.7688
Avellana Translation	31.72	0.5707	0.3841	0.3042	0.7621

Table 5: Comparisons on Task 1 and 3 across teams (ranked via SACREBLEU chosen by the organisers).

Task-1, and using METEOR/ROUGE for Task-3 among all teams who submitted on-time before the shared task deadline.

We looked into the translation outputs from clinic-NLLB for error analysis, and it shows that some of the translation errors come from very literal translation, and others come from gender related mistakes. In conclusion, *our two stage experimental investigations verify our hypothesis and RQs from different aspects*.

We also doubt if the official automatic metrics used for ClinSpEn challenge can correctly distinguish the NMT systems because mostly they do not really measure the translation output quality but the similarity to the gold standard single reference. Therefore, domain specific automatic evaluation metrics or metrics better measuring semantic similarities might be needed.

In the future work, we plan to carry out more experimental investigations from qualitative aspects looking into translation errors using human experts and classifying them into possible categories with examples and statistics, especially from clinical domain. This will allow us to validate automatic metrics with professional human judgements for this domain.

We will continue to fine-tune our models towards different domains and languages and use more of the available corpus for current clinical domain challenge task. We also plan to try different state-of-the-art pre-trained language models for evaluation.

Acknowledgements

The authors thank the ClinSpEn2022 shared task organisers for preparing the data set and evaluation platforms, thank Darryl Estrada for in communication with us during the competition. The authors thank the project support from HIPS (R126035 task A05) and from JigSaw (R124782 task A07) at The University of Manchester. We thank the open research projects Marian Helsinki NMT and Meta-AI's wmt21.dense-24-wide.En-X (2021) and NLLB (2022) we used.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, OndÅ™ej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina EspaÃ±a-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (wmt21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. *CoRR*, abs/1907.10902.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Matthias Aßenmacher. 2021. *Comparability, Evaluation and Benchmarking of large pre-trained language models*. Ph.D. thesis, Ludwig Maximilian University of Munich.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL*.

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9347–9359, Online. Association for Computational Linguistics.

Souradip Chakraborty, Ekaba Bisong, Shweta Bhatt, Thomas Wagner, Riley Elliott, and Francesco Mosconi. 2020. BioMedBERT: A pre-trained biomedical language model for QA and IR. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 669–679, Barcelona, Spain (Online). International Committee on Computational Linguistics.

KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259.

Franck Dernoncourt, Ji Young Lee, and Peter Szolovits. 2017. NeuroNER: an easy-to-use program for named-entity recognition based on neural networks. *Conference on Empirical Methods on Natural Language Processing (EMNLP)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Chenhe Dong, Guangrun Wang, Hang Xu, Jiefeng Peng, Xiaozhe Ren, and Xiaodan Liang. 2021. EfficientBERT: Progressively searching multilayer perceptron via warm-up knowledge distillation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1424–1437, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gleb Erofeev, Irina Sorokina, Lifeng Han, and Serge Gladkoff. 2021. cushLEPOR uses LABSE distilled knowledge to improve correlation with human translations. In *Proceedings of Machine Translation Summit XVIII: Users and Providers Track*, pages 421–439, Virtual. Association for Machine Translation in the Americas.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, Errors, and Context: A Large-Scale Study of Human Evaluation for Machine Translation. *arXiv e-prints*, page arXiv:2104.14478.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021. Results of the wmt21 metrics shared task: Evaluating metrics with expert-based human evaluations on ted and news domain. In *Proceedings of the Six Conference on Machine Translation*. Association for Computational Linguistics.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna Clinciu, Dipanjan Das, Kaustubh D.

Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Rubungo Andre Niyongabo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM Benchmark: Natural Language Generation, its Evaluation and Metrics. *arXiv e-prints*, page arXiv:2102.01672.

Yvette Graham, Timothy Baldwin, and Nitika Mathur. 2015. Accurate evaluation of segment-level machine translation metrics. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1183–1191.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthcare*, 3(1).

Aaron L. F. Han, Derek F. Wong, and Lidia S. Chao. 2012. LEPOR: A robust evaluation metric for machine translation with augmented factors. In *Proceedings of COLING 2012: Posters*, pages 441–450, Mumbai, India. The COLING 2012 Organizing Committee.

Aaron Li-Feng Han, Derek F. Wong, Lidia S. Chao, Yi Lu, Liangye He, Yiming Wang, and Jiaji Zhou. 2013a. A description of tunable machine translation evaluation systems in WMT13 metrics task. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 414–421, Sofia, Bulgaria. Association for Computational Linguistics.

Lifeng Han. 2014. *LEPOR: An Augmented Machine Translation Evaluation Metric*. University of Macau, MSc. Thesis.

Lifeng Han. 2022a. *An investigation into multi-word expressions in machine translation*. Ph.D. thesis, Dublin City University.

Lifeng Han. 2022b. An overview on machine translation evaluation. *arXiv preprint arXiv:2202.11027*.

Lifeng Han, Valerio Antonini, Ghada Alfattni, Alfredo Madrid, Warren Del-Pinto, Judith Andrew, William G. Dixon, Meghna Jani, Ana Maria Aldana, Robyn Hamilton, Karim Webb, and Goran Nenadic. 2022a. A transformer-based machine learning framework using conditional random fields as decoder for clinical text mining. In *HealTAC 2022: the 5th Healthcare Text Analytics Conference. Poster 19.*

Lifeng Han, Gleb Erofeev, Irina Sorokina, Serge Gladkoff, and Goran Nenadic. 2022b. Using massive multilingual pre-trained language models towards real zero-shot neural machine translation in clinical domain.

Lifeng Han and Serge Gladkoff. 2022. Meta-evaluation of translation evaluation methods: a systematic up-to-date overview. In *Tutorial at LREC2022*, Marseille, France.

Lifeng Han, Gareth Jones, and Alan Smeaton. 2020. AlphaMWE: Construction of multilingual parallel corpora with MWE annotations. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 44–57, online. Association for Computational Linguistics.

Lifeng Han, Gareth Jones, Alan Smeaton, and Paolo Bolzoni. 2021a. Chinese character decomposition for neural MT with multi-word expressions. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 336–344, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Lifeng Han, Irina Sorokina, Gleb Erofeev, and Serge Gladkoff. 2021b. cushLEPOR: customising hLEPOR metric using optuna for higher agreement with human judgments or pre-trained language model LaBSE. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1014–1023, Online. Association for Computational Linguistics.

Lifeng Han, Derek F. Wong, Lidia S. Chao, Liangye He, Yi Lu, Junwen Xing, and Xiaodong Zeng. 2013b. Language-independent model for machine translation evaluation with reinforced factors. In *Machine Translation Summit XIV*, pages 215–222. International Association for Machine Translation.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, USA. Association for Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang.

2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Zeyang Liu, Ke Zhou, and Max L. Wilson. 2021. Meta-evaluation of Conversational Search Evaluation Metrics. *arXiv e-prints*, page arXiv:2104.13453.

Matouš Macháček and Ondřej Bojar. 2013. Results of the WMT13 metrics shared task. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 45–51, Sofia, Bulgaria. Association for Computational Linguistics.

Shaimaa Marzouk. 2021. An in-depth analysis of the individual impact of controlled language rules on machine translation output: a mixed-methods approach. *Machine Translation*.

Aurélie Névéol, Antonio Jimeno Yepes, Mariana Neves, and Karin Verspoor. 2018. Parallel corpora for the biomedical domain. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adrien Pavao, Isabelle Guyon, Anne-Catherine Letournel, Xavier Baró, Hugo Escalante, Sergio Escalera, Tyler Thomas, and Zhen Xu. 2022. Codalab competitions: An open source platform to organize scientific challenges. *Technical report*.

Laura Perez-Beltrachini and Mirella Lapata. 2021. Models and datasets for cross-lingual summarisation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9408–9423, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Alv Romell and Jacob Curman. 2022. *Multilingual Large Scale Text Classification for Automotive Troubleshooting Management*. MSc Thesis, Lund University.

Sebastian Ruder. 2021. Recent Advances in Language Model Fine-tuning. `http://ruder.io/recent-advances-lm-fine-tuning`.

Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Graham Neubig, and Melvin Johnson. 2021. XTREME-R: Towards More Challenging and Nuanced Multilingual Evaluation. *arXiv e-prints*, page arXiv:2104.07412.

Viktor Schlegel. 2021. *Emerging evaluation paradigms in natural language understanding: a case study in machine reading comprehension*. Ph.D. thesis, University of Manchester.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218. Citeseer.

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. Facebook ai's wmt21 news translation task submission. In *Proc. of WMT*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Conference on Neural Information Processing System*, pages 6000–6010.

Marta Villegas, Ander Intxaurrondo, Aitor Gonzalez-Agirre, Montserrat Marimon, and Martin Krallinger. 2018. The mespen resource for english-spanish medical machine translation and terminologies: census of parallel corpora, glossaries and term translations. *LREC MultilingualBIO: multilingual biomedical text processing*.

Lingzhi Wang, Huang Hu, Lei Sha, Can Xu, Kam-Fai Wong, and Daxin Jiang. 2021. Finetuning large-scale pre-trained language models for conversational recommendation with knowledge graph. *CoRR*, abs/2110.07477.

Yuping Wu, Lifeng Han, Valerio Antonini, and Goran Nenadic. 2022. On cross-domain pre-trained language models for clinical text mining: How do they perform on data-constrained fine-tuning? In *Forthcoming*.

Appendix

More training logs from clinic-Marian:

- global step = 3516
- training loss = 1.2236216656855212
- train runtime = 1945.9989
- train samples per second = 115.622
- trian steps per second = 1.807
- total flos = 29470348636323840

Parameters reported by SACREBLEU:

- lowercase = Ture
- tokenize = 13a

Summer: WeChat Neural Machine Translation Systems for the WMT22 Biomedical Translation Task

Ernan Li, Fandong Meng and and Jie Zhou
Pattern Recognition Center, WeChat AI, Tencent Inc, China
{cardli,fandongmeng,withtomzhou}@tencent.com

Abstract

This paper introduces WeChat's participation in WMT 2022 shared biomedical translation task on Chinese→English. Our systems are based on the Transformer(Vaswani et al., 2017), and use several different Transformer structures to improve the quality of translation. In our experiments, we employ data filtering, data generation, several variants of Transformer, fine-tuning and model ensemble. Our Chinese→English system, named Summer, achieves the highest BLEU score among all submissions.

1 Introduction

This article describes the WeChat's participation in WMT 2022 shared biomedical translation task on Chinese→English. We improve the translation quality of the system by increasing the diversity of model structure and data, fine-tuning the model with in-domain data, inserting tags at the beginning of each source sentence and selecting models with high diversity and good performance for ensemble.

For model architectures, our system adopt BIG and DEEP Transformer models which contain 10-layer and 20-layer encoders, 10240 and 4096 filter sizes, respectively, with TRANSFORMER-BIG setting (Vaswani et al., 2017). In order to increase the diversity of the model, we use structures such as Average Attention Transformer (AAN) (Zhang et al., 2018) and Mixed-AAN Transformer architecture (Zeng et al., 2021) in the decoder part.

For data generation, we use back-translation (Sennrich et al., 2016a), knowledge distillation (Kim and Rush, 2016), and forward-translation (Zeng et al., 2021) to improve data quality. And we use some data augmentation methods to improve the model robustness, such as adding synthetic noise and dynamic top-p sampling (Zeng et al., 2021). Furthermore, according to the different sources of the corpora, we add tags at the beginning of the source sentence to perform domain adaptation.

For fine-tuning, we use in-domain bilingual corpus to fine-tune models from the general domain to the biomedical domain, and use target denoising (Meng et al., 2020) to improve the diversity of models and mitigate training-generation discrepancy.

For model ensemble, we use Self-BLEU (Zhu et al., 2018) to evaluate the similarity between models. We take the prediction of one model as the reference and use the prediction of the other model to calculate the BLEU score. The higher the Self-BLEU score, the lower the diversity of the models.

In the remainder of this paper, we start with presenting the data strategy in Section 2. Then we describe our system details in Section 3. Section 4 presents the experimental results. Finally, we conclude our work in Section 5.

2 Data

In this section, we introduce the details of bilingual and monolingual data used in this shared task.

2.1 Bilingual Corpus

Our baseline model is trained with out-of-domain (OOD) data from WMT 2022 shared task on general machine translation[1]. Additionally, we use in-house data (depicted in Table 1 as OOD-IN-HOUSE) to improve performance of baseline model. With regard to in-domain data, firstly, we use the in-domain bilingual corpus provided by the WMT 2022 shared biomedical translation task[2] (depicted in Table 1 as IND-BIO). And we use the Champollion[3] tool to align the sentences in the corpus. Then, we collect in-domain Chinese→English

[1] https://statmt.org/wmt22/translation-task.html

[2] https://github.com/biomedical-translation-corpora/corpora

[3] http://champollion.sourceforge.net/

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 920 - 924
December 7-8, 2022 ©2022 Association for Computational Linguistics

(depicted in Table 1 as IND-TAUS) sentence pairs from TAUS[4].

2.2 Monolingual Corpus

The out-of-domain monolingual corpora are collected from WMT 2022 shared task on general machine translation and the in-house monolingual data. With regard to in-domain data, the English part of the bilingual corpus in other languages provided by the WMT 2022 shared biomedical translation task is used as in-domain monolingual data.

3 System overview

In this section, we introduce the details of our system used in the WMT 2022 shared biomedical translation task. Our system adopts data filtering, data generation, model architectures, fine-tuning and ensemble.

3.1 Data Filtering

For data filtering, we use the following rules for bilingual corpus:

- Normalize punctuation with Moses scripts on both English and Chinese.
- Filter out sentence pairs that are the same at the source and target.
- Filter out sentence pairs whose source sentence's language recognition result is different from the original language.
- Filter out sentence pairs with a source-to-target length ratio greater than 1:3.
- Filter out the sentences longer than 150 words or exceed 40 characters in a single word.

Besides these rules, we use fast-align[5] to filter out the sentence pairs with low alignment scores. We also filter out sentence pairs in which English sentences contain Chinese characters.

3.2 Data Generation

In this section, we introduce the approaches of data generation in our system, including back-translation, knowledge distillation, forward-translation, synthetic noise and tagging.

3.2.1 Back-Translation

Back-translation (Hoang et al., 2018) is the most commonly used data augmentation method in neural machine translation. Following the previous work (Edunov et al., 2018), we use following strategies to generate back translations to improve the diversity the training data:

- Beam search: We use beam search to generate the pseudo corpus with beam size setting to 4.
- Dynamic top-p sampling: Following the work (Zeng et al., 2021), at each decoding step, we select a word from the smallest set whose cumulative probability exceeds p, with p varying from 0.9 to 0.95 during the data generation process.

3.2.2 Knowledge Distillation

For knowledge distillation (Kim and Rush, 2016; Wang et al., 2021), we use the corpus generated from the teacher models to train the student models.

3.2.3 Forward-Translation

For forward-translation, we use an ensemble model to generate forward translations with the source-language monolingual corpus as input.

3.2.4 Synthetic Noise

For synthetic noise, we add different noises at the source side of the pseudo corpus to improve the diversity of the data and improve the robustness of the model:

- Randomly replace some source tokens with $< unk >$.
- Randomly delete some tokens from the source sentence.
- Randomly swap the two tokens in the source sentence in the specify window.

3.2.5 Tagging

For tagging, inspired by (Johnson et al., 2017), we insert a tag at the beginning of each source sentence to denote its type: $< BT >$ for the back-translation data, $< NOISE >$ for the synthetic noise data, $< REAL >$ for the ground-truth bilingual corpus and $< FT >$ for the forward-translation data. Furthermore, we insert a tag at the second position of each sentence to denote its domain: $< BIO >$ for the in-domain data, $< NEWS >$ for the data from WMT22 general

[4] https://taus-corona-corpus.s3.amazonaws.com/en-zh.txt.gz

[5] https://github.com/clab/fast_align

LANGUAGE	OOD-NEWS	OOD-IN-HOUSE	IND-BIO	IND-TAUS
bilingual corpus	30.6M	90M	89K	0.4M
monolingual corpus	220M	50M	6.9M	–

Table 1: Data used for training the system, where *OOD-NEWS* is the out-of-domain data provided by WMT22 general translation task. *OOD-IN-HOUSE* is the out-of-domain data collected from in-house corpus. *IND-BIO* is the in-domain data provided by WMT22 shared biomedical translation task. And *IND-TAUS* is the in-domain data collected manually (not from MEDLINE, as depicted in 2.1). *M* denotes *million* and *K* denotes *thousand*.

translation task and $< INHOUSE >$ for the data from our in-house corpus. At inference time, we always use the $< REAL >$ and $< BIO >$ tag.

3.3 Model Architectures

In this section, we introduce the model architectures used by our system, including Transformer (Big/Deep), Average Attention Transformer (AAN) and Mixed Average Attention Transformer (Mixed-AAN) (Zeng et al., 2021).

3.3.1 Transformer

Our baseline models are Big- and Deep-Transformer (Vaswani et al., 2017) models. In our experiments, we use multiple model configurations with 20-layer and 30-layer encoders for deep models and 10-layers encoders for big models, and use 6-layers decoders for all models. The hidden size is set to 1024 and the filtering size is set from 4096 to 10240.

3.3.2 Average Attention Transformer

To increase the diversity between models, we adopt Average Attention Transformer (Zhang et al., 2018), where the average attention is used to replace self-attention in the decoder. AAN summarizes the historical information of previous positions by means of cumulative average, which increases diversity with almost no harm to the quality of the model.

3.3.3 Mixed-AAN Transformers

Following the previous work (Zeng et al., 2021), we adopt the Mixed-AAN Transformers to further improve the diversity and quality of models. In this experiment, we only use two architectures of Mixed-AAN:

- Self-first: In the decoder part, we use self-attention as the first layer, and then use average attention and self-attention alternately.

- AAN-first: In the decoder part, we use average attention as the first layer, and then use self-attention and average attention alternately.

3.4 Fine-tuning

For fine-tuning, we mainly use the in-domain data provided by WMT22 shared biomedical translation task for domain adaption (Luong and Manning, 2015; Li et al., 2019). In order to prevent the model from overfitting, as well as to improve the diversity of the model after domain transfer, we adopt target denoising (Meng et al., 2020). We add synthetic noise at the decoder inputs during fine-tuning. Therefore, with target denoising, the model becomes more robust. The method of adding synthetic noise is described in Section 3.2.4.

3.5 Ensemble

After obtaining a variety of different models through the above methods, we need to find the best model combination to get the best result. In general, the better the model performance and the greater the diversity between models, the better the performance for the model ensemble. To measure diversity, we use Self-BLEU (Zhu et al., 2018) to evaluate the similarity between models. Overall, we select 6 models from 52 candidate models for ensemble. All the candidate models are generated by different combinations of data and different training strategies as described earlier.

4 Experiments

4.1 Settings

Our experiment is based on Fairseq [6]. The single models are carried out on 8 NVIDIA V100 / A100 GPUs. We adopt the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.998$. The batch-size is set to 4096 tokens, and the “update-freq" is set to 4, and the warmup step is set to 4000 and the learning rate is set to 0.0005.

4.2 Pre-processing and Post-processing

The Chinese sentences are segmented by a in-house segmentation tool and English sentences are seg-

[6] https://github.com/pytorch/fairseq

System	BLEU
Baseline	34.57
+ IND-TAUS	35.65
+ IND-BIO	40.96
+ OOD-IN-HOUSE	41.88
+ Back-Translation	42.8
+ Knowledge Distillation	43.12
+ Forward-Translation	43.32
+ Multi BT	44.11
+ Finetune	44.96
+ Target denoise finetune	45.1
Baseline_TAG	34.48
+ IND-TAUS	35.62
+ IND-BIO	41.07
+ OOD-IN-HOUSE	42.14
+ Back-Translation	43.91
+ Knowledge Distillation	44.14
+ Forward-Translation	44.39
+ Multi BT	45.23
+ Finetune	45.43
+ Target denoise finetune	45.54
+ Ensemble	**46.91**⋆

Table 2: Translation performance on WMT21 biomedical translation task testset. ⋆ is the system we submitted. Multi BT means the iterative back-translation (Hoang et al., 2018) which use with different part of data and different generation strategies.

mented by the tokenizer toolkit in Moses[7]. We normalize punctuation using Moses scripts on both English and Chinese. For handling uppercase and lowercase of the English letters, we add a special token at the beginning of a word to denote uppercase (_UU_) and title case (_U_). By this way to reduce the size of the word list and reduce the difficulty of model training. For instance, *"We are together NOW."* → *"_U_ we are together _UU_ now."*. We use BPE (Sennrich et al., 2016b) with 32K operations for all the languages.

With the regard of post-processing, we use *detokenizer.perl* on the English translations provided in Moses.

4.3 Results

The experimental results of Chinese→English on WMT21 OK-aligned biomedical test set are shown in Table 2.

Compared with the baseline model (Baseline_TAG), the in-domain bilingual data (+IND-BIO) provided by WMT22 shared biomedical translation task brings a huge improvement, with 6.5 point increase in BLEU score. After adding the in-house out-of-domain corpus (+OOD-IN-HOUSE), we further gain +1.1 BLEU. We further obtain +1.8 BLEU by applying back-translation (+Back-Translat), and +0.23 BLEU by using knowledge distillation (+Knowledge Distillation), and +0.25 BLEU by using forward-translation (+ Forward-Transla). After using iterative back-translation (Hoang et al., 2018) (+Multi BT) described in Table 2, we further achieve improvement of +0.84 BLEU.

Additionally, we can find that the model with TAG was similar to the model without TAG in early stage experiments. As the number of data categories and data domains increases, the model with tags gradually demonstrates its advantages. Our best single model (+Target denoise fine) achieves 45.54 BLEU score, and we finally achieve 46.91 BLEU score by model ensemble (+Ensemble).

[7] http://www.statmt.org/moses/

5 Conclusion

We introduce WeChat's participation in WMT 2022 shared biomedical translation task on Chinese→English. Our system is based on the Transformer (Vaswani et al., 2017), and uses several different Transformer structures such as Average Attention and Mixed-AAN to improve the performance. We use several data augmentation methods such as iterative back-translation, knowledge distillation, forward-translation and synthetic noise. We use tags to assist the model in domain learning and use in-domain fine-tuning with target denoising to domain transfer. Finally a Self-BLEU based ensemble method is used for model ensemble. Overall, our system achieves 46.91 BLEU score on WMT21 OK-aligned biomedical test set, and we achieve the highest BLEU score among all submissions.

References

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The NiuTrans machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266, Florence, Italy.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79.

Fandong Meng, Jianhao Yan, Yijin Liu, Yuan Gao, Xianfeng Zeng, Qinsong Zeng, Peng Li, Ming Chen, Jie Zhou, Sifan Liu, and Hao Zhou. 2020. WeChat neural machine translation systems for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 239–247, Online.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Fusheng Wang, Jianhao Yan, Fandong Meng, and Jie Zhou. 2021. Selective knowledge distillation for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6456–6466, Online.

Xianfeng Zeng, Yijin Liu, Ernan Li, Qiu Ran, Fandong Meng, Peng Li, Jinan Xu, and Jie Zhou. 2021. WeChat neural machine translation systems for WMT21. In *Proceedings of the Sixth Conference on Machine Translation*, pages 243–254, Online. Association for Computational Linguistics.

Biao Zhang, Deyi Xiong, and Jinsong Su. 2018. Accelerating neural transformer via an average attention network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, Melbourne, Australia.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.

Optum's Submission to WMT22 Biomedical translation tasks

Sahil Manchanda
sahil_manchanda@optum.com

Saurabh Bhagwat
saurabh_bhagwat@optum.com

Abstract

This paper describes Optum's submission to the Biomedical Translation task of the seventh conference on Machine Translation (WMT22). The task aims at promoting the development and evaluation of machine translation systems in their ability to handle challenging domain-specific biomedical data. We made submissions to two sub-tracks of ClinSpEn 2022, namely, ClinSpEn-CC (clinical cases) and ClinSpEn-OC (ontology concepts). These sub-tasks aim to test translation from English to Spanish. Our approach involves fine-tuning a pre-trained transformer model using in-house clinical domain data and the biomedical data provided by WMT. The fine-tuned model results in a test BLEU score of 38.12 in the ClinSpEn-CC (clinical cases) subtask, which is a gain of 1.23 BLEU compared to the pre-trained model.

1 Introduction

The quality of Neural Machine Translation (NMT) was boosted by the use of Recurrent Neural Networks (RNN) for machine translation. In this approach, the source sentence is fed to an encoder which outputs a context vector. This context vector is fed to the decoder to output the target language text (Cho et al., 2014). Some approaches also use Long Short Term Memory (Hochreiter and Schmidhuber, 1997) for this task (Sutskever et al., 2014).

Machine Translation (MT) systems after seeing great progress in recent years have been found to be sensitive to synthetic and natural noise in input, distributional shift, and adversarial examples (Koehn and Knowles, 2017; Belinkov and Bisk, 2017; Durrani et al., 2019; Anastasopoulos et al., 2019; Michel et al., 2019). Fine-tuning has proven to be a successful technique to carry out this task. One of the most prominent variations is described in (Chu and Wang, 2018), which trains an NMT model on out-of-domain corpora until model convergence and then resumes training from step 1 on a mix of in-domain and out-of-domain data.

A fine-grained human evaluation research of the transformer based systems and state-of-the-art recurrent systems was carried out on the translation from English to Chinese. The evalution results shows reduction in errors by 31 percent and significantly less errors in 10 out of 22 error categories when using Transformer based MT systems. (Ye and Toral, 2020). Another research has shown that improved efficiency and accuracy can be obtained by converting a pre-trained transformer into its efficient recurrent counterpart. A swap procedure is implemented which replaces softmax attention of a pertained transformer with its linear-complexity recurrent alternative followed by fine-tuning. Fine-tuning has proven to help reduce the training cost and improve efficiency and accuracy (Kasai et al., 2021).

We took part in WMT 2022 Biomedical translation task from English to Spanish using the fine-tuning approach on the Transformer based models and we describe our efforts in this paper. The paper is structured as follows. The data sets and their preparation is outlined in Section 3 and Section 4, followed by details of the experiments carried out and their results in Section 5. We then present the summary of our findings and conclusion in Section 6.

2 Related Work

Machine translation systems out of domain performance has been negatively impacted to the extent that they completely sacrifice adequacy for the sake of fluency. Hence, the presence of domain inconsistency is considered a key challenge in machine translation (Koehn and Knowles, 2017). The common approach to tackle this challenge is firstly to train an MT system on a (generic) source domain and secondly to fine-tune it on a (specific) target domain (Luong and Manning, 2015; Freitag and Al-Onaizan, 2016; Servan et al., 2016; Chu et al., 2017), followed by continuous fine-tuning of data

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 925 - 929
December 7-8, 2022 ©2022 Association for Computational Linguistics

sets which are similar to the target domain (Sajjad et al., 2017), or to dynamically change the balance of data towards the target domain (van der Wees et al., 2017). An alternative approach is to train systems on multiple domains simultaneously, while adding domain-specific tags to the input examples (Kobus et al., 2016).

Other methods include the works around Dual Contextual (DC) module, which is an extension of the conventional self-attention unit, to effectively make use of both, local and global contextual information. This work aims to further improve the sentence representation ability of the encoder and decoder sub-networks, thus enhancing the overall performance of the translation model (Ampomah et al., 2021). Domain adaptation methods include instance weighing, data selection (Wang et al., 2017) and incorporating a domain classifier (Chen et al., 2017; Britz et al., 2017).

Some language pairs do not have enough parallel text for training. Hence, to counter the data sparsity problem of the NMT training some have used various strategies like augmenting training data, exploiting training data from other languages, alternative learning strategies that use only monolingual data (Haque et al., 2021). Some of the researchers have made use of monolingual data available either in the target domain, for example, by training the decoder on these data sets (Domhan and Hieber, 2017), or by back-translating (Sennrich et al., 2016), or in the source domain, using similar techniques (Zhang and Zong, 2016).

3 Data

In the experiments described in this paper, we use data sets from both the general and clinical domains. ParaCrawl, EMEA, and WMT are available in the public domain, while, M&R Letters is a data set internal to Optum. The M&R in-domain data set comprises of medical claim correspondence letters sent to the insurance customers which have been manually translated to Spanish. Among the public data sets, ParaCrawl is the largest publicly available parallel corpora for European languages. EMEA is a multi-lingual parallel corpus made out of PDF documents from the European Medicines Agency. We have used data from all three subtracks namely, clinical cases, clinical terminology, and ontology concepts of the ClinSpEn data set provided by WMT. Table 1 summarizes the data sets used and their size. It is important to note that we generate train and test splits on ParaCrawl (general domain), EMEA, and M&R data sets (clinical domain) and evaluate on these. For WMT, we use all 8K sentence pairs as training data and share evaluation BLEU scores computed by WMT submission system on their hidden test set.

Data Fragment	Sentences	Domain
ParaCrawl	38M	General
M&R Letters	492K	Medical
EMEA	15K	Clinical
WMT	8K	Clinical

Table 1: Data sets used in this work and corresponding source and number of sentences in each.

4 Data Preparation

The Data preparation very closely follows the steps outlined in (Manchanda and Grunin, 2020). The additional steps are listed below.

1. **Language Check Elimination**
   Sentences not from the intended language were eliminated.

2. **Length difference check**
   The internal data that we used comprised of correspondence letters to our customers anonymized and their manual translations. It was found upon a close observation that manual translations differ depending on the translator. Sometimes, the same phrase can be translated multiple ways or some additional information can be added unintentionally to the translation which can confuse the learning algorithm leading to under-fitting. We eliminated any translation that differs from the source sentence in length by more than 40 percent.

5 Experiments and Results

As described in the Data Section (3), We are using data sets from both the General domain and Medical/Clinical domain. To fine-tune the model, we have a 2-GPU setup with a docker container deployed on on-premise machines containing all the required packages to fine-tune the OPUS en-es translation model [1]. We use HuggingFace transformers library (Wolf et al., 2020) for all our experiments.

[1] https://huggingface.co/Helsinki-NLP/opus-mt-en-es

The following fine-tuning experiments are done on the transformer model used by the Helsinki-NLP opus-mt-en-es model. As evident from its model card, this model was trained on general-purpose English to Spanish training corpus and in these experiments, we will try to fine-tune the model to the clinical domain.

Since the data provided by the sub-task was limited, we used the entire WMT 2022 data as training data and used train-test splits on other clinical domain data sets to test the success of fine-tuning.

One of our key observation while doing the experiments and serving these models on production systems were that they regularly need to be checked for over-fitting and hallucination errors. In addition to evaluation by BLEU scores, We do a "**Sanity check**" by running an inference with source language strings of various lengths to mimic handwritten text and check if the translation is not adding extra tokens.

1. **Experiment 0: Reference Baseline**
   We use the model already pre-trained without any fine-tuning as our reference baseline and compare our fine-tuning results against this to determine the better models.

2. **Experiment 1: Mix of General and In-domain data**
   First, we fine-tune the general purpose model on a mix of in-domain and public data set. Our in-domain data sets are M&R correspondence letters, EMEA clinical data set and WMT 2022. We mix these with 2 million sentences randomly selected from the ParaCrawl corpus to keep the model from over-fitting to only one domain. We keep the learning rate on the higher side (1e-5) for this experiment and train for 1 epoch only. We do not add length difference check (2) in this experiment on the in-domain data.

3. **Experiment 2: Fine-tuning on only In-domain data**
   Our next experiment was to fine-tune the public model on only the in-domain data sets. This experiment contains all the data preparation steps. The learning rate for this experiment was kept lower as compared to the previous experiment (1e-6) as the data was purely in-domain.

Figure 1 shows a graph of the BLEU scores at evaluation time for all the above-mentioned experiments. Along with the BLEU scores on the test splits of general and Clinical (EMEA/M&R) datasets, this figure also shows the test BLEU scores provided by WMT on their hidden test sets. We observe that the model trained on only general-purpose data (Experiment 0) performs decently on both in-domain and general-purpose data sets. Experiments 1 and 2 yield better results on the EMEA/M&R data sets, and degrade a little on the general-purpose data sets. It can be noted that both experiments have the same scores on general and EMEA/M&R datasets. This indicates that the approach of fine-tuning with a high learning rate with some general domain data present (experiment 1) and fine-tuning with a low learning rate only on the in-domain data (experiment 2) yields very similar results.

However, Experiment 2 yields the best results on the WMT test data set and hence is our primary submission to the task. It is interesting to note that the gain on the BLEU scores of EMEA and M&R datasets is more significant as compared to the gain in WMT BLEU scores. One of the major reasons for that could be the amount of data available for this particular domain.

6 Conclusion

We fine-tuned a publicly available model in multiple ways using different combinations of data from various sources. We showed how fine-tuning is sensitive to new domains and can show promising results if done diligently. This paper shows the results of fine-tuning on a single domain but we think that fine-tuning on any new domain would provide gains in the translation quality. The scale of this gain, however, can depend on the amount of training data available in that particular domain.

7 Limitation

As evident from our experiments and results, in-domain machine translation involves some trade-off in translation quality amongst domains. When we tried to fine-tune a translation model to a new domain, the BLEU scores on the general domain drop. The users of the fine-tuned model need to be cognizant of the fact that while these models are the best for the domain they were fine-tuned for, they might not be the best to translate general handwritten text which lacks the structure of the fine-tuning data. We recommend separate specialized models for different use cases.

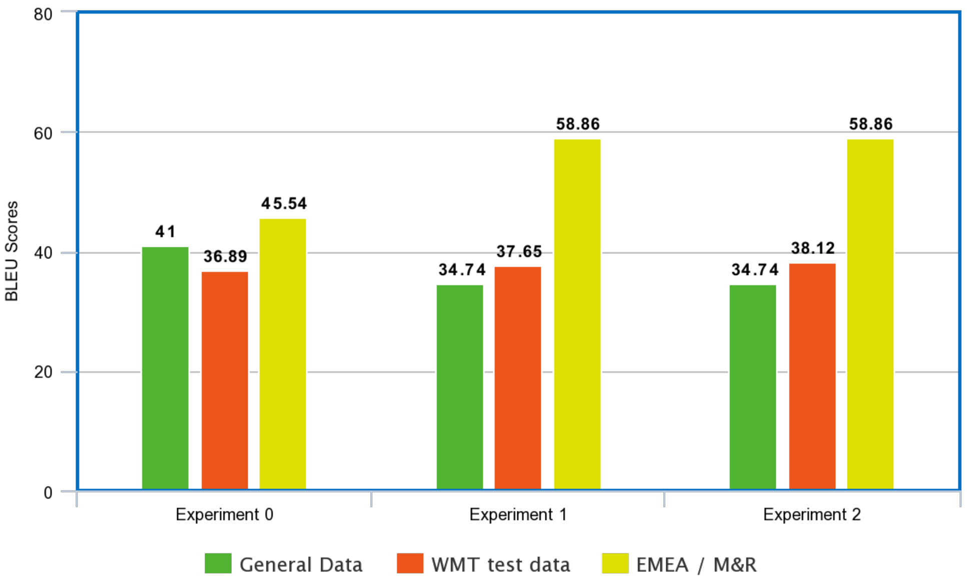


Figure 1: BLEU scores on evaluation data sets

References

Isaac Kojo Essel Ampomah, Sally McClean, and Glenn Hawe. 2021. Dual contextual module for neural machine translation. *Machine Translation*, 35(4):571–593.

Antonios Anastasopoulos, Alison Lui, Toan Q. Nguyen, and David Chiang. 2019. Neural machine translation of text from non-native speakers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3070–3080, Minneapolis, Minnesota. Association for Computational Linguistics.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *CoRR*, abs/1711.02173.

Denny Britz, Quoc Le, and Reid Pryzant. 2017. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 118–126, Copenhagen, Denmark. Association for Computational Linguistics.

Boxing Chen, Colin Cherry, George Foster, and Samuel Larkin. 2017. Cost weighting for neural machine translation domain adaptation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 40–46, Vancouver. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of simple domain adaptation methods for neural machine translation. *CoRR*, abs/1701.03214.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tobias Domhan and Felix Hieber. 2017. Using target-side monolingual data for neural machine translation through multi-task learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1500–1505, Copenhagen, Denmark. Association for Computational Linguistics.

Nadir Durrani, Fahim Dalvi, Hassan Sajjad, Yonatan Belinkov, and Preslav Nakov. 2019. One size does not fit all: Comparing NMT representations of different granularities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1504–1516, Minneapolis, Minnesota. Association for Computational Linguistics.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *CoRR*, abs/1612.06897.

Rejwanul Haque, Chao-Hong Liu, and Andy Way. 2021. Recent advances of low-resource neural machine translation. *Machine Translation*, 35.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Jungo Kasai, Hao Peng, Yizhe Zhang, Dani Yogatama, Gabriel Ilharco, Nikolaos Pappas, Yi Mao, Weizhu Chen, and Noah A. Smith. 2021. Fine-tuning pretrained transformers into rnns. *CoRR*, abs/2103.13076.

Catherine Kobus, Josep Maria Crego, and Jean Senellart. 2016. Domain control for neural machine translation. *CoRR*, abs/1612.06140.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *CoRR*, abs/1706.03872.

Minh-Thang Luong and Christopher Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 76–79, Da Nang, Vietnam.

Sahil Manchanda and Galina Grunin. 2020. Domain informed neural machine translation: Developing translation services for healthcare enterprise. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 255–261, Lisboa, Portugal. European Association for Machine Translation.

Paul Michel, Xian Li, Graham Neubig, and Juan Pino. 2019. On evaluation of adversarial perturbations for sequence-to-sequence models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3103–3114, Minneapolis, Minnesota. Association for Computational Linguistics.

Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Yonatan Belinkov, and Stephan Vogel. 2017. Neural machine translation training in a multi-domain scenario. In *Proceedings of the 14th International Conference on Spoken Language Translation*, pages 66–73, Tokyo, Japan. International Workshop on Spoken Language Translation.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Christophe Servan, Josep Maria Crego, and Jean Senellart. 2016. Domain specialization: a post-training domain adaptation for neural machine translation. *CoRR*, abs/1612.06141.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark. Association for Computational Linguistics.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017. Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1488, Copenhagen, Denmark. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yuying Ye and Antonio Toral. 2020. Fine-grained human evaluation of transformer and recurrent approaches to neural machine translation for english-to-chinese. *CoRR*, abs/2006.08297.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Austin, Texas. Association for Computational Linguistics.

Huawei BabelTar NMT at WMT22 Biomedical Translation Task: How we further improve domain-specific NMT

Weixuan Wang , Xupeng Meng, Suqing Yan, Ye Tian, Wei Peng[*]
Artificial Intelligence Application Research Center, Huawei Technologies
peng.wei1@huawei.com

Abstract

This paper describes Huawei Artificial Intelligence Application Research Center's neural machine translation system ("BabelTar"). Our submission to the WMT22 biomedical translation shared task covers language directions between English and the other seven languages (French, German, Italian, Spanish, Portuguese, Russian, and Chinese). During the past four years, our participation in this domain-specific track has witnessed a paradigm shift of methodology from a purely data-driven focus to embracing diversified techniques, including pre-trained multilingual NMT models, homograph disambiguation, ensemble learning, and preprocessing methods. We illustrate practical insights and measured performance improvements relating to how we further improve our domain-specific NMT system.

1 Introduction

The existing mainstream neural machine translation (NMT) system is predominantly data-driven. Our participation in WMT biomedical tasks traced back from 2019 has witnessed pursuits extending beyond this modality. In our WMT20 and WMT21 submissions, various domain adaption technologies (Bawden et al., 2020; Akhbardeh et al., 2021) have been applied including practical approaches fine-tuning on general-purpose models, back-translation (Sennrich et al., 2016a) and leveraging in-domain dictionaries (Peng et al., 2020; Wang et al., 2021). Despite achieving state-of-the-art (SOTA) BLEU scores for most of our submissions in the last two years, under-translation occurred in the "English ↔ Chinese" due to the models' incapability to handle long sentences (Wang et al., 2021). It was rectified by ensembling the affected model with the baseline, resulting in a decrease in BLEU scores. In addition, the models trained predominately with the general domain data still face challenges associated with domain adaptation.

In this paper, we present practical insights into how we further improve Huawei Artificial Intelligence Application Research Center's neural machine translation system ("BabelTar") in domain-specific machine translation. This year, our participation in the WMT22 biomedical translation task covers language directions between "English (EN)" and the other seven languages "German (DE)", "Spanish (ES)", "French (FR)", "Italian (IT)", "Portuguese (PT)", "Russian (RU)" and "Chinese (ZH)". More specifically, we adopt in-house general-purposed bilingual NMT models built upon the transformer-big architecture (Vaswani et al., 2017) and a pre-trained multilingual NMT model (M2M100) (Fan et al., 2021) with an M2M100-418M configuration as baseline models. Finetuned with the in-domain data provided by the organizer, the back-translated monolingual Medline data in English dating before July 2018, the in-domain dictionaries enhanced with terminologies, the models can be improved significantly over the last year's submissions, for example, +1.18 BLEU on "EN → IT" and +1.24 BLEU on "EN → DE". Leveraging the knowledge learned in addressing the ambiguities caused by homographs, we can further boost +0.65 BLEU in the language direction of "EN → ZH". By optimizing the sequence length during decoding, we successfully solve the issue of under-translation in the language pair of "EN ↔ ZH".

2 The Data

In this section we detail the bilingual and monolingual corpora used in this shared task (Table 1).

- **OOD**: The general domain data (OOD) are in-house data used to train the baseline models.
- **IND**: In all directions, we use the in-domain

[*] Corresponding author

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 930 - 935
December 7-8, 2022 ©2022 Association for Computational Linguistics

Directions	**Train**					**Dev.**	**Test**	**Vocab.**
	OOD	**IND**	**IND-Dict.**	**IND-Aug.**	**IND-BT.**			
EN→DE	6M	2.4M	62.5K	-	5.5M	1.1K	340	42K
DE→EN	6M	2.4M	62.5K	-	53M	1.1K	370	42K
EN→ES	3.3M	1.1M	131K	-	-	1K	410	40K
ES→EN	3.3M	1.1M	131K	-	52.5M	1K	382	40K
EN→FR	3M	2.8M	62.5K	-	-	1.6K	342	40K
FR→EN	3M	2.8M	62.5K	889K	53M	1.6K	314	40K
EN→IT	6M	139K	60.6k	235K	695k	0.8K	339	40K
IT→EN	6M	139K	60.6k	235K	55M	0.8K	327	40K
EN→PT	3M	7.1M	60.3K	-	-	1k	403	32K
PT→EN	3M	7.1M	60.3K	-	52.5M	1k	423	32K
EN→RU	3M	32K	60.4K	-	-	792	161	40K
RU→EN	3M	32K	60.4K	-	52.5M	792	210	40K
EN→ZH	3M	-	60.1K	847K	-	5K	347	50K
ZH→EN	3M	-	60.1K	847K	-	5K	311	50K

Table 1: Data used for training and evaluating the system. “M” is the acronym for “million”, and K stands for “thousand”, indicating the records of sentences, lexicon pairs or vocabularies. The Dev. datasets are extracted from the training datasets, and we use WMT21 shared task test data to evaluate our submission this year.

data (IND) provided by the shared task organizers to finetune the baseline models. [1] The IND data consists of WMT-released bitexts from Pubmed, UFAL, [2] Medline, [3] MeSpEn, [4] Scielo [5] and Brazilian Thesis and Dissertations.[6]

- **IND-dict.**: The lexicon pairs are collected from SNOMED-CT, [7] DOPPS[8] and WFOT.[9] Other terminologies are from Babel linguistics, [10] with COVID-19 related terms obtained from Neulab. [11]

- **IND-Aug.**: We augment the in-domain data using parallel corpora collected from TAUS [12] for the English ↔ Spanish, English ↔ French, English ↔ Italian, and English ↔ Chinese language pairs.

- **IND-BT.**: A batch of monolingual Medline data in English (IND-BT.) dated before July 2018 has been collected and back-translated for data augmentation. The official released IND data from WMT is also back-translated. The models used for back-translation are from our last year’s shared task (Wang et al., 2021).

It is noted that OOD, IND, IND-dict. and IND-Aug. are combined and subsequently partitioned for training and evaluation.

3 The Approaches

The proposed systems are finetuned using the following methods. It is noted that bilingual models are trained on one Tesla V100 GPU, taking approximately 8-20 hours. All multilingual models are trained on eight Tesla V100 GPUs, taking 6-50 hours, depending on the volumes of data involved.

3.1 Multilingual NMT Models

Unlike our previous submissions focusing merely on bilingual NMT models, we leverage pre-trained multilingual NMT models (M2M-100) in the shared task this year.

[1] http://www.statmt.org/wmt21/biomedical-translation-task.html

[2] https://ufal.mff.cuni.cz/ufal_medical_corpus

[3] https://github.com/biomedical-translation-corpora/corpora

[4] https://temu.bsc.es/mespen/

[5] https://figshare.com/articles/dataset/A_Large_Parallel_Corpus_of_Full-Text_Scientific_Articles/5382757

[6] https://figshare.com/articles/A_Parallel_Corpus_of_Thesis_and_Dissertations_Abstracts/5995519

[7] https://www.nlm.nih.gov/healthit/snomedct/index.html

[8] https://static.lexicool.com/dictionary/XJ9XO98314.pdf

[9] https://static.lexicool.com/dictionary/HY1TK12777.pdf

[10] https://babel-linguistics.com/resources/glossaries/

[11] https://github.com/neulab/covid19-datashare/tree/master/parallel/terminologies

[12] https://md.taus.net/corona

System	EN→DE	EN→ES	EN→FR	EN→IT	EN→PT	EN→RU	EN→ZH
Bi-baseline	31.25	51.01	**47.27**	43.92	48.94	32.26	39.98
Bi-best	**32.49**	**51.81**	47.27	**45.10**	53.87	34.41	**42.23**
Multi-baseline	21.46	42.13	36.31	33.53	38.73	25.25	24.04
Multi-best	30.5	51.48	45.5	43.46	**53.98**	**37.14**	38.69
WMT22 Submission	33.42	44.75	37.85	48.48	52.55	37.03	47.68
Official Best	39.14	52.35	40.17	48.48	52.55	41.27	55.71

System	DE→EN	ES→EN	FR→EN	IT→EN	PT→EN	RU→EN	ZH→EN
Bi-baseline	40.46	50.79	48.82	44.73	47.36	44.69	**39.62**
Bi-best	**41.57**	**53.47**	**48.86**	44.73	**59.41**	47.69	39.62
Multi-baseline	33.67	43.23	35.73	36.43	41.84	39.76	21.57
Multi-best	40.68	52.02	46.37	**45.67**	58.08	**48.48**	34.96
WMT22 Submission	43.75	59.02	49.36	49.89	56.03	46.75	46.12
Official Best	46.95	60.45	50.95	49.89	56.03	50.01	46.17

Table 2: BLEU scores on related submissions. The Bi-baseline models represent the best bilingual models in our WMT21 participation (Wang et al., 2021) for language pairs in EN ↔ DE, EN ↔ FR, EN ↔ IT and EN ↔ ZH with others are out-of-domain bilingual NMT models newly trained for EN ↔ ES, EN ↔ PT and EN ↔ RU. The results of the Multi-baseline are the pre-trained multilingual NMT models from M2M100-418M on related language directions. The Bi-best and Multi-best are the bilingual and multilingual NMT models trained using the depicted methods achieving the best results.

Data	EN→IT	IT→EN	EN→PT	PT→EN	EN→RU	RU→EN
Baseline	33.53	36.43	38.73	41.84	25.25	39.76
+IND	42.17	43.72	50.12	54.74	36.25	47.09
+IND-all + IND	**43.46 (+1.29)**	**45.67 (+1.95)**	**53.98 (+3.86)**	**58.08 (+3.34)**	**37.14 (+0.89)**	**48.48 (+1.39)**

Table 3: Effects of applying different finetuning order to train English⇔Italian, English⇔Portuguese, English⇔Russian M2M-100 models on WMT21.

3.2 Domain-specific Dictionaries

Leveraging domain-specific dictionaries is proved a viable solution for domain adaptation in NMT (Peng et al., 2020; Wang et al., 2021) to enhance IND data coverage. A terminology dictionary is generated from the collected lexicons and attached to the end of the parallel corpus for each language direction to train the models.

3.3 Ensemble Learning

Ensemble learning is a representative method aggregating several models' predictions to obtain more accurate predictions. We average the probabilities of NMT output layers at each time step as depicted in Garmash and Monz (2016). In these experiments, we choose the top 3 best bilingual NMT models to participate in ensemble learning.

3.4 Homograph Disambiguation

Homographs may confuse an NMT model in selecting an inaccurate prediction due to conflicting word sense meanings in different domains. We design a novel approach to tackle homographic issues of NMT in the latent space to handle cross-domain ambiguities. The method is under review and will appear in another venue.

3.5 Preprocessing and Postprocessing

The under-translation problem presented in Wang et al. (2021) is associated with the inability of an NMT model to handle long sentences. The presence of noisy training data may cause under-translation. We optimize the preprocessing pipeline to include techniques like sentence segmentation, punctuation normalization, special tokens replacement, etc., leading to a resolution of the under-

Models	EN→DE	DE→EN	EN→FR	FR→EN	EN→IT	IT→EN	EN→ZH	ZH→EN
Model-1	31.25	40.46	**47.27**	48.82	43.92	**44.73**	**42.23**	**39.62**
Model-2	31.65	40.42	47.21	48.34	43.92	44.22	41.58	39.14
Model-3	31.01	40.17	47.25	48.55	45.04	44.05	41.29	38.92
Ensemble	**32.49**	**41.57**	46.79	**48.86**	**45.10**	44.71	41.36	38.50

Table 4: Results from the ensemble learning of the top three models on WMT21.

translation problem. More specifically, we first perform punctuation normalization to standardize data formats using Moses library (Koehn et al., 2007). Sentencepiece approach (Sennrich et al., 2016b) is subsequently used to tokenize the sentences into a series of subwords. Sentences with a length longer than a threshold (i.e., 80 subwords) are segmented to handle issues wrt under-translation. Preprocessing also replaces some unique tokens with placeholders, such as roman numbers, to avoid the out-of-vocabulary (OOV) problem. Postprocessing strategies are used to recover the previously segmented sentences. The detokenization is performed to convert subwords into words. Finally, we apply specific rules to handle punctuations and remove undesirable spaces.

4 Experimental Results and Analysis

As OOD data also contribute to the domain-specific NMT (Wang et al., 2021), both OOD data and IND data are used to finetune the NMT bilingual and multilingual NMT models. OK-aligned WMT21 test data are used for evaluation in the experiments. The BLEU scores are evaluated using the MTEVAL script from Moses (Koehn et al., 2007) with results shown in Table 2.

4.1 Multilingual NMT

It is challenging to finetune a pre-trained multilingual NMT model with hundreds of millions of parameters (i.e., 418 millions parameters for M2M-100-418M) with limited numbers of in-domain data. We design a two-stage training procedure in which a multilingual baseline initially finetuned on IND data of all available language pairs (“IND-all”) is subsequently trained on data from a specific language pair (“IND”). As depicted in Table 3, such a two-stage training method (“IND-all + IND”) is more effective than a simple finetuning step, achieving a significant improvement to the BLEU score (up to +3.86). Multilingual NMT models outperform bilingual NMT models, particularly for low-resource language pairs, such as EN ↔ RU and IT → EN (shown in Table 2).

4.2 Ensemble Decoding

We choose the three best models to ensemble in all experiments, including our best model submitted in the WMT21 shared task and the other two models trained following the methods depicted in this paper. Unlike the way mentioned in Wang et al. (2021) in averaging the logarithmic probabilities of a decoded token, we average the outputs of the output layer. This proves to be a more effective approach than the one used in previous years’ submissions. The results are shown in Table 4. We have not investigated means to ensemble a pre-trained multilingual NMT model with our SOTA bilingual NMT models due to time and resource constraints in this year’s shared task.

4.3 The Effect of Homograph Disambiguation

Table 6 demonstrates the effectiveness of applying a method designated for homographic disambiguation. It can be observed that resolving homographic issues in domain-specific NMT can significantly improve the BLEU score to up to +0.65.

4.4 Preprocessing to Solve Under-translation

To handle issues relating to under-translation, we design a segmentation strategy to break sentences longer than 80 subwords. Combined with other preprocessing techniques, we can further improve the performance of our domain-specific NMT system. Table 7 shows a +0.89 BLEU enhancement. A comparison of translated examples is shown in Table 5 to aid our understanding.

5 Discussion

It is the fourth year we have participated in this shared task, and we have made significant progress in our submissions measured against officially released test data from previous years. But the improvements for some language directions are not always accompanied by a consistent uplift of BLEU

Sentence	Example
Input	The disease duration ranged from 2 weeks to 60 months (median, 4 months), and the affected segment was C All the patients were followed up 3 to 42 months (median, 12 months).
Wang et al. (2021)	病程2周
This year	病程2周-60个月（中位，4个月），累及节段为C。随访3-42个月（中位，12个月）。
Input	The median age of the 30 patients was 56.5 (28-80) years old, among them, 25 patients were primary plasma cell leukemia, and 5 patients were secondary plasma cell leukemia.
Wang et al. (2021)	30例患者的中位年龄为56.5（28
This year	30例患者中位年龄为56.5（28-80）岁，其中原发性浆细胞白血病25例，继发性浆细胞白血病5例。

Table 5: A comparison of examples produced by Wang et al. (2021) and by models submitted this year in the translation task for EN → ZH.

Model	EN→ZH
Baseline	41.58
Homographic Disambiguation	**42.23 (+0.65)**

Table 6: The effect of applying an approach designed for homograph disambiguation to domain-specific NMT. The baseline is the NMT model for EN ⇔ ZH, without the assistance of the homograph disambiguation technique.

Model	EN→ZH
Baseline	40.69
Preprocessing + Baseline	**41.58 (+0.89)**

Table 7: Compared results between models with or without preprocessing when training EN → ZH translation model on WMT21.

for the contest year. The learned NMT models still suffer from "out of distribution" issues many deep learning models have encountered. Apart from maintaining the NMT models with a large amount of the latest IND data, we need to design deep learning systems to adapt to changes in distributions (Bengio et al., 2021).

On another point, we realized that the reference data sometimes do not reflect the ground truth of the translation during our manual evaluation process. It raises a related question about the rationale of using BLEU as an exclusive automatic evaluation criterion. Although BLEU may remain the default metric for evaluating machine translation quality, we strongly suggest the community investigate complementary metrics capable of accommodating good translation results with semantics variations in this shared task.

6 Conclusion

This paper depicts Huawei's neural machine translation system ("BebelTar") and the submission to the WMT22 biomedical shared task. The submission consists of fourteen models covering language directions between English and all seven other languages available in this track. We can improve the domain-specific NMT significantly by leveraging a broad range of techniques, which includes pretrained multilingual NMT models, lexicon-based enhancement, homograph disambiguation, ensemble learning, preprocessing and postprocessing, etc. In the meantime, we share practical insights on achieving the measured performance, hoping to contribute to the machine translation community in this shared task. Our future work will focus on investigating mechanisms to adapt a domain-specific NMT model to different distributions.

Acknowledgements

We express our gratitude to colleagues from HUAWEI Artificial Intelligence Application Center for their continuous support. We also appreciate the WMT22 organizers for hosting this shared task and anonymous reviewers' insightful comments, which enable us to further improve domain-specific NMT.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Rachel Bawden, Giorgio Maria Di Nunzio, Cristian Grozea, Inigo Jauregi Unanue, Antonio Jimeno-Yepes, Nancy Mah, David Martínez, Aurélie Névéol, Mariana L. Neves, Maite Oronoz, Olatz Perez-de-Viñaspre, Massimo Piccardi, Roland Roller, Amy Siu, Philippe Thomas, Federica Vezzani, Maika Vicente Navarro, Dina Wiemann, and Lana Yeganova. 2020. Findings of the WMT 2020 biomedical translation shared task: Basque, italian and russian as new additional languages. In *Proceedings of the Fifth Conference on Machine Translation, WMT@EMNLP 2020, Online, November 19-20, 2020*, pages 660–687. Association for Computational Linguistics.

Yoshua Bengio, Yann Lecun, and Geoffrey Hinton. 2021. Deep learning for ai. *Communications of the ACM*, 64(7):58–65.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22:107:1–107:48.

Ekaterina Garmash and Christof Monz. 2016. Ensemble learning for multi-source neural machine translation. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 1409–1418. ACL.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Wei Peng, Jianfeng Liu, Minghan Wang, Liangyou Li, Xupeng Meng, Hao Yang, and Qun Liu. 2020. Huawei's submissions to the WMT20 biomedical translation task. In *Proceedings of the Fifth Conference on Machine Translation, WMT@EMNLP 2020, Online, November 19-20, 2020*, pages 857–861. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Weixuan Wang, Wei Peng, Xupeng Meng, and Qun Liu. 2021. Huawei aarc's submissions to the wmt21 biomedical translation task: Domain adaption from a practical perspective. In *Proceedings of the Sixth Conference on Machine Translation*, pages 868–873, Online. Association for Computational Linguistics.

HW-TSC Translation Systems for the WMT22 Biomedical Translation Task

Zhanglin Wu, Jinlong Yang, Zhiqiang Rao, Zhengzhe Yu, Daimeng Wei, Xiaoyu Chen, Zongyao Li, Hengchao Shang, Shaojun Li, Ming Zhu, Yuanchang Luo, Yuhao Xie, Miaomiao Ma, Ting Zhu, Lizhi Lei, Song Peng, Hao Yang, Ying Qin
Huawei Translation Service Center, Beijing, China
{wuzhanglin2,yangjinlong7,raozhiqiang,yuzhengzhe,weidaimeng,chenxiaoyu35, lizongyao,shanghengchao,lishaojun18,zhuming47,luoyuanchang,xieyuhao2, mamiaomiao,zhuting20,leilizhi,pengsong2,yanghao30,qinying}@huawei.com

Abstract

This paper describes the translation systems trained by Huawei translation services center (HW-TSC) for the WMT22 biomedical translation task in five language pairs: English↔German (en↔de), English↔French (en↔fr), English↔Chinese (en↔zh), English↔Russian (en↔ru) and Spanish→English (es→en). Our primary systems are built on deep Transformer with a large filter size. We also utilize R-Drop, data diversification, forward translation, back translation, data selection, finetuning and ensemble to improve the system performance. According to the official evaluation results in OCELoT[1] or CodaLab[2], our unconstrained systems in en→de, de→en, en→fr, fr→en, en→zh and es→en (clinical terminology sub-track) get the highest BLEU scores among all submissions for the WMT22 biomedical translation task.

1 Introduction

Machine translation (MT) refers to the automatic translation of text from one language to another, and the biomedical translation task aims to evaluate the performance of MT systems in the biomedical domain. In this year's biomedical translation task, our team (HW-TSC) participates in five language pairs, including en↔de, en↔fr, en↔zh, en↔ru and es→en (clinical terminology sub-track).

Since the size of in-domain (ID) data is limited, we first use a large amount of out-of-domain (OOD) data to train our baseline neural machine translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2015; Gehring et al., 2017; Vaswani et al., 2017) system, which is a deep transformer model (Dou et al., 2018; Li et al., 2019) leveraging R-Drop (Wu et al., 2021) training strategy. We then use the collected ID data (except the data from medical database) to further train the NMT model for domain transfer. To better use the limited ID training data, we employ data selection to extract ID data from OOD data, in addition to basic data augmentation strategies including data diversity, forward translation and back translation. Finally, we use finetuning (Dakwale and Monz, 2017) and model ensemble (Wang et al., 2020b) to further improve model performance in the biomedical domain.

This paper is structured as follows: we describe data size and data pre-processing methods in section 2; the model structure and training methods in section 3; final results in section 4; and conclusion in section 5.

2 Dataset

2.1 Data volume

The data size for each language pair for the WMT22 biomedical translation task is shown in Table 1. The OOD bilingual data, used to train our baseline model, comes from the WMT general MT task and our internal corpus; while the ID bilingual and monolingual data, used for transferring the domain (Yang et al., 2021), come from Biomedical Translation, UFAL Medical Corpus and our internal corpus. As there is no ID monolingual data, we use the OOD monolingual instead.

2.2 Data Pre-processing

The data pre-processing process is as follows:

- Remove duplicate sentences (Khayrallah and Koehn, 2018; Ott et al., 2018).
- Remove sentences with mismatched parentheses and quotation marks.
- Filter out sentences of which punctuation percentage exceeds 0.4.
- Filter out sentences with the character-to-word ratio greater than 12 or less than 1.5.

[1] https://ocelot-wmt22.mteval.org

[2] https://codalab.lisn.upsaclay.fr/competitions/6696#results

	bilingual					monolingual				
	en↔de	en↔fr	en↔zh	en↔ru	es→en	en	de	fr	zh	ru
OOD	200M	600M	200M	200M	200M	-	10M	-	-	40M
ID	2.75M	6.05M	10.87M	0.24M	8.1M	46M	-	2M	92M	-

Table 1: The data size for each language pair in the WMT22 Biomedical Translation Task

- Filter out sentences with more than 150 words.
- Apply langid (Joulin et al., 2017, 2016) to filter sentences in other languages.
- Use fast-align (Dyer et al., 2013) to filter out sentence pairs that are poorly aligned.

It should be noted that for en↔de, en↔fr, en↔ru and es→en translation task, we adopt joint SentencePiece model (SPM) (Kudo and Richardson, 2018; Kudo, 2018) for word segmentation, with a vocabulary of 32k. As for en↔zh translation task, we use Jieba tokenizer[3] to pre-segment Chinese sentences, and Moses tokenizer (Koehn et al., 2007) to pre-segment English sentences. Then we use joint Byte Pair Encoding (BPE) (Sennrich et al., 2016) to perform subword segmentation on Chinese and English sentences. The vocabulary size of BPE is also set to 32k.

3 System Overview

3.1 Model

Transformer (Vaswani et al., 2017), as the current mainstream architecture for NMT, adopts a fully self-attention mechanism, which can realize algorithm parallelism, speed up model training, and improve model performance. Deep Transformer, as an improvement of Transformer, increases the number of encoder layers and uses pre-layer-normalization to further improve model performance. Therefore, for all four language pairs, we adopt the Deep Transformer (Wei et al., 2021) model architecture: Based on the Transformer-big model architecture, our Deep Transformer model features pre-layer-normalization, 25-layer encoder, 6-layer decoder, 16-head self-attention, 1024-dimension word embedding and 4096-dimension hidden state.

3.2 R-Drop

Dropout (Srivastava et al., 2014) is a powerful and widely used technique for regularizing deep neural networks. Though it can help improve training effectiveness, the randomness introduced by dropouts may lead to inconsistencies between training and inference. R-Drop (Wu et al., 2021) forces the output distributions of different sub-models generated by dropout be consistent with each other. Therefore, we use R-Drop to augment the baseline model for each task and reduce inconsistencies between training and inference.

3.3 Data Diversification

Data diversification (Nguyen et al., 2020) is a simple and effective strategy to improve the performance of NMT. It uses predictions from multiple forward and backward models, and combines the results with the original data to train the final NMT model. The method does not require additional monolingual data and is applicable to all NMT models. It is more efficient than knowledge distillation (Wang et al., 2021) and dual learning (He et al., 2016). In our en↔de, en↔fr, en↔zh and en↔ru translation tasks, we use only a forward model and a backward model to generate synthetic data, and then mix the synthetic data with the bilingual data for NMT model training.

3.4 Forward Translation

Forward translation (Wu et al., 2019), also known as self-training (Imamura and Sumita, 2018), refers to using a forward NMT model to translate source-side monolingual data to generate synthetic bilingual data, which is then used to expand the training data size. Forward translation usually relies on beam search (Freitag and Al-Onaizan, 2017) decoding to generate synthetic data. Therefore, we adopt the forward translation method based on beam search decoding.

3.5 Back Translation

Back translation (Sennrich et al., 2015; Edunov et al., 2018) refers to translating the target monolingual data back to the source language, and then using the synthetic data to increase the training data size. This method has been proven effective in improving the NMT model performance. There are many back translation methods, among which sampling (Graça et al., 2019), noise (Edunov et al.,

[3] https://github.com/fxsjy/jieba

2018) or tagged (Caswell et al.) back-translation methods work better. In the scenario where forward translation and back translation are used in combination (Wu et al., 2019), the improvement effect brought by sampling back translation is more significant. In our translation task, we adopt sampling back translation method.

3.6 Data Selection

Data selection (van der Wees et al., 2017) is a data augmentation method that we use to select ID bilingual data from OOD bilingual data. Inspired by the domain feature calculation in curriculum learning (Wang et al., 2020a), we use an ID NMT model and an OOD NMT model to calculate the decoding probability of OOD bilingual data. The bilingual data of which ID decoding probability is higher than OOD decoding probability can be selected as additional ID data. The data selection process is also shown in Algorithm 1:

Algorithm 1: Data selection process

```
Input  : ID NMT model $\theta_I$, OOD NMT model $\theta_O$ and OOD bilingual data set $D_O$.
Output : ID bilingual data set $D_I$.
for each sentence pair $(x, y) \in D_O$ do
    // $x$ is the source sentence, $y$ is the target sentence.
    $score = \frac{\log P(y|x;\theta_I) - \log P(y|x;\theta_O)}{|y|}$
    if $score > 0$ then
        add $(x, y)$ to $D_I$
    end
end
```

3.7 Finetuning

Finetuning (Dakwale and Monz, 2017) is a way to achieve domain transfer. In our translation task, we adopt a two-stage finetuning strategy. In the first stage, we use ID bilingual data to continue training the OOD NMT model, and then use the data augmentation strategy mentioned above to improve the model performance. In the second stage, we use the development set and synthetic data generated from the source-side text in the test set to finetune the ID model for more fine-grained domain transfer.

3.8 Ensemble

Ensemble (Wang et al., 2020b) is a widely used method to integrate different models for better performance. It is worth noting that when using ensemble, increasing the number of models does not always lead to better performance, and sometimes even causes performance deterioration. Therefore, for each track, we train four models on the same data, and go through all combinations of models to choose the one that performs best on the development set. This is also the model selection strategy (Yang et al., 2021) we use in the WMT21 biomedical translation task.

4 Experimental Result

During the training phase, we use Pytorch-based Fairseq[4] (Ott et al., 2019) open-source framework, and use deep Transformer model architecture as our benchmark system. Each model is trained using 8 GPUs with a batch size of 2048. The update frequency is 4 and the learning rate is 5e-4, the label smoothing rate (Szegedy et al., 2016) is 0.1, the warm-up steps is 4000, and the dropout is 0.3. Adam optimizer (Kingma and Ba, 2015) with β1=0.9 and β2=0.98 is also used. Furthermore, we use reg_label_smoothed_cross_entropy as the loss function and set reg-alpha to 5 when applying R-Drop (Wu et al., 2021) training strategy. In the evaluation phase, we use Marian[5] (Junczys-Dowmunt et al., 2018) for decoding and then calculate the sacrebleu[6] (Post, 2018) on the WMT21 OK-aligned biomedical test set to measure the performance of each model.

4.1 en↔de

For en↔de track, Table 2 shows the results of using the methods mentioned above to improve the model performance. The results show that continuing training with ID bilingual data on the basis of an OOD baseline improves en→de translation performance by 1.6 BLEU, but has little effect on the de→en track, with an increase of only 0.1 BLEU. Data selection significantly improves en↔de translation performance by 0.9-1.2 BLEU. In addition, other training strategies also bring small performance improvements.

4.2 en↔fr

Table 3 shows the results of en↔fr model. The results show that data diversity brings the greatest improvement to translation of both directions

[4] https://github.com/facebookresearch/fairseq
[5] https://github.com/marian-nmt/marian
[6] https://github.com/mjpost/sacrebleu

System	en→de	de→en
OOD R-Drop baseline	27.3	39.7
+ ID bilingual data continue training	28.9	39.8
+ data diversification	29.0	40.1
+ forward translation & back translation	29.4	41.3
+ data selection	30.3	42.5
+ dev set & synthetic test set finetuning	30.8	42.9
+ ensemble	**31.0**	**43.2**

Table 2: BLEU scores of en↔de on the WMT21 OK-aligned biomedical test set.

System	en→fr	fr→en
OOD R-Drop baseline	44.8	46.1
+ ID bilingual data continue training	45.3	46.3
+ data diversification	46.0	47.6
+ forward translation & back translation	46.3	47.7
+ data selection	-	47.8
+ dev set & synthetic test set finetuning	46.8	48.4
+ ensemble	**46.9**	**48.6**

Table 3: BLEU scores of en↔fr on the WMT21 OK-aligned biomedical test set.

(0.7 BLEU and 1.3 BLEU respectively). However, data selection has little impact on fr→en translation, and even no impact on en→fr translation. We assume this is because not much ID bilingual data is selected from the OOD data.

4.3 en↔zh

For en↔zh track, continuing training with ID bilingual data on the basis of an ODD baseline, as well as data diversity, bring the greatest impact on the model performance, while data selection has the least impact. In addition, the methods such as forward translation & back translation, dev set & synthetic test set finetuning and ensemble have little improvement on en→zh translation, but have a great improvement on zh→en translation. The detailed results of en↔zh translation are shown in Table 4.

4.4 en↔ru

As shown in Table 5, for the en↔ru track, the results are similar to en↔zh translation task. Continuing training with ID bilingual data and data diversity have the greatest impact on model performance, while data selection does not lead to performance improvement. In addition, the performance improvements brought by other methods are also relatively limited.

4.5 es→en

We also participate in the es→en clinical terminology sub-track (ClinSpEn-CT) this year. The sample set contains 7,000 terms that are extracted from medical literature and clinical records, with a particular focus on diseases, symptoms, findings, etc. The translations are generated and revised by professional medical translators. We extract 1000 sentences from the sample set as the dev set.

The results are shown in Table 6. All chrF and BLEU scores are calculated on this dev set. Unlike other experiments above, for es→en clinical terminology sub-task, we abandon forward translation method for the sake of maintaining terminology accuracy. Instead, we perform two rounds of back translation using monolingual English ID data. Finally, we finetune the model with 6000 bilingual terms, which results in a significant improvement on the dev set.

System	en→zh	zh→en
OOD R-Drop baseline	38.5	32.1
+ ID bilingual data continue training	41.4	35.0
+ data diversification	42.5	36.4
+ forward translation & back translation	42.7	37.3
+ data selection	42.8	-
+ dev set & synthetic test set finetuning	43.0	38.7
+ ensemble	**43.1**	**39.3**

Table 4: BLEU scores of en↔zh on the WMT21 OK-aligned biomedical test set.

System	en→ru	ru→en
OOD R-Drop baseline	35.4	46.8
+ ID bilingual data continue training	41.0	48.9
+ data diversification	41.7	50.3
+ forward translation & back translation	42.3	50.4
+ data selection	-	-
+ dev set & synthetic test set finetuning	42.4	50.9
+ ensemble	**42.5**	**51.1**

Table 5: BLEU scores of en↔ru on the WMT21 OK-aligned biomedical test set.

4.6 Results In OCELoT Or CodaLab

The BLEU scores of our submissions to the WMT22 Biomedical Translation Task on OCELoT and CodaLab (ClinSpEn-CT) are shown in Table

System	chrF	BLEU
OOD R-Drop baseline	0.76	49.5
+ ID bilingual data continue training	0.77	50.7
+ back translation	0.79	53.4
+ 2nd round back translation	0.79	54.1
+ 6000 bilingual terms finetuning	0.82	56.7
+ ensemble	**0.82**	**57.2**

Table 6: chrF (Popović, 2015) and BLEU scores of es→en on the WMT22 biomedical ClinSpEn-CT 1000 sample set.

	en→de	de→en	en→fr	fr→en	en→zh	zh→en	en→ru	ru→en	es→en
our submission system	**38.7**	**45.6**	**38.8**	**48.6**	**49.9**	43.0	43.3	50.3	**41.57**

Table 7: BLEU scores of our submission systems on WMT22 Biomedical Translation Task on OCELoT or CodaLab, where the highest BLEU scores among all submissions are bolded.

7, where our submitted systems achieve the highest BLEU scores in six language directions of the WMT22 biomedical translation task. In conclusion, from the results on the WMT21 OK-aligned biomedical test set, continuing training with ID bilingual data, data diversity, forward translation and back translation have great impacts on the NMT model performance. When the OOD bilingual data contains a certain amount of ID bilingual, the data selection method can also achieve a good boost effect. In addition, dev set & synthetic test set finetuning and ensemble can lead to further performance gains.

5 Conclusion

This paper presents our translation system for the WMT22 en↔de, en↔fr, en↔zh, en↔ru and es →en biomedical translation task. During the experiment, we use R-Drop and ID bilingual data finetuning methods to build our ID translation system, and then use data diversity, forward translation, back translation and data selection methods to expand the size of training data for training a better system. We also adopt finetuning and ensemble to further improve the system performance. According to the official evaluation results in OCELoT or CodaLab, our submitted systems achieve the highest BLEU scores in six language directions of the WMT22 biomedical translation task.

References

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Isaac Caswell, Ciprian Chelba, and David Grangier. Tagged back-translation. *WMT 2019*, page 53.

P Dakwale and C Monz. 2017. Fine-tuning for neural machine translation with limited degradation across in-and out-of-domain data.

Zi-Yi Dou, Zhaopeng Tu, Xing Wang, Shuming Shi, and Tong Zhang. 2018. Exploiting deep representations for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4253–4262.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 56–60.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252.

Miguel Graça, Yunsu Kim, Julian Schamper, Shahram Khadivi, and Hermann Ney. 2019. Generalizing back-translation in neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 45–52.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 820–828.

Kenji Imamura and Eiichiro Sumita. 2018. Nict self-training approach to neural machine translation at nmt-2018. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 110–115.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Herve Jegou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models.

Armand Joulin, Édouard Grave, Piotr Bojanowski, and Tomáš Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, et al. 2018.

Marian: Fast neural machine translation in c++. In *ACL (4)*.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83.

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *ACL (1)*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP (Demonstration)*.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, et al. 2019. The niutrans machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266.

Xuan-Phi Nguyen, Shafiq Joty, Kui Wu, and Ai Ti Aw. 2020. Data diversification: A simple strategy for neural machine translation. *Advances in Neural Information Processing Systems*, 33:10018–10029.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *ICML*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*, pages 3104–3112.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826. IEEE.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Fusheng Wang, Jianhao Yan, Fandong Meng, and Jie Zhou. 2021. Selective knowledge distillation for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6456–6466.

Wei Wang, Ye Tian, Jiquan Ngiam, Yinfei Yang, Isaac Caswell, and Zarana Parekh. 2020a. Learning a multi-domain curriculum for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7711–7723.

Yiren Wang, Lijun Wu, Yingce Xia, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2020b. Transductive ensemble learning for neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6291–6298.

Daimeng Wei, Zongyao Li, Zhanglin Wu, Zhengzhe Yu, Xiaoyu Chen, Hengchao Shang, Jiaxin Guo, Minghan Wang, Lizhi Lei, Min Zhang, et al. 2021. Hwtsc's participation in the wmt 2021 news translation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 225–231.

Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. 2021. R-drop: Regularized dropout for neural networks. *Advances in Neural Information Processing Systems*, 34:10890–10905.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4207–4216.

Hao Yang, Zhanglin Wu, Zhengzhe Yu, Xiaoyu Chen, Daimeng Wei, Zongyao Li, Hengchao Shang, Minghan Wang, Jiaxin Guo, Lizhi Lei, et al. 2021. Hwtsc's submissions to the wmt21 biomedical translation task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 879–884.

Unbabel-IST at the WMT Chat Translation Shared Task

João Alves[*1], **Pedro Henrique Martins**[*1,3,4],
José G. C. de Souza[1], **M. Amin Farajian**[1], **André F. T. Martins**[1,3,4]
[1]Unbabel, Lisbon, Portugal, [2]INESC-ID, Lisbon, Portugal
[3]Instituto de Telecomunicações, Lisbon, Portugal
[4]Instituto Superior Técnico, University of Lisbon, Portugal

Abstract

We present the joint contribution of IST and Unbabel to the WMT 2022 Chat Translation Shared Task. We participated in all six language directions (English ↔ German, English ↔ French, English ↔ Brazilian Portuguese). We addressed the lack of domain-specific data with a lightweight adaptation approach, using mBART50, a large pretrained language model trained on millions of sentence-pairs, as our base model. We fine-tune it using a two-step fine-tuning process. In the first step, we fine-tune the model on publicly available data. In the second step, we use the validation set. After having a domain-specific model, we explore the use of kNN-MT as a way of incorporating domain-specific data at decoding time.[1]

1 Introduction

In recent years, neural machine translation (NMT) has seen remarkable advances due to the increasingly powerful models (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017). The translation of conversational text is an important and challenging application for machine translation, specially in the customer support domain, since international companies have an increasing need to offer customer support in various languages. However, this domain has not been substantially explored in machine translation research.

In the Chat Translation shared task, the goal is to understand the context's impact in conversational text translation, and to study the feasibility of multilingual systems for customer support translation. This year, the focus was on the case in which we have a centralizing costumer support with English speaking agents and a translation layer between agent and costumer, allowing the communication with customers which speak different languages.

In this paper we discuss our submission to this task. Our submitted system covers all 3 language pairs: English-German, English-Brazilian Portuguese, and English-French, in both directions: we translate the agent utterance from English to the other language and the customer utterances from the other language to English. As no training data is provided for this task, we recur to the use of the pre-trained multilingual machine translation model mBART50 (§2.1; Tang et al. (2020)) and perform domain adaptation through fine-tuning (§2.2) with domain-specific data and by retrieving similar examples from domain-specific datastores (§2.3). To increase the size of training examples that can be used to fine-tune the model and to create the domain-specific datastore we search for similar examples on publicly available datasets (§3.1) and perform back-translation of the provided monolingual data (§3.2).

2 Models

In this section, we describe the model that we used to tackle this shared task. We start by describing the base model. Then, we describe the techniques used to adapt the base model to customer support chat translation.

2.1 Base Model

As our base model, we use the mBART50 (Tang et al., 2020) "one-to-many" (English to 49 other languages) or "many-to-one" (49 languages to English), depending on the language direction. mBART50 can translate sentences between English and 49 different languages, which include the languages present in this shared task (German, French and Brazilian Portuguese). It consists of a pre-trained encoder-decoder transformer that is first pretrained on a auto-denoising task with monolingual data from 25 languages (mBART; Liu et al. (2020)) and then further pre-trained on an extended set of monolingual data that comprises

[*] Equal contribution.

[1] The code was based on: `https://github.com/deep-spin/efficient_kNN_MT`.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 943 - 948
December 7-8, 2022 ©2022 Association for Computational Linguistics

50 languages. Then, to adapt the model to perform machine translation, Tang et al. (2020) performed multilingual fine-tuning on machine translation, using data from the 50 supported languages. For this, they used three different configurations: "one-to-many", "many-to-one", and "many-to-many". The first two are obtained by fine-tuning the model with the bilingual data, having English as the source or target language, respectively. The latter is obtained by fine-tuning the model with all the language pairs combinations (using English as the pivot language to obtain the bilingual data).

2.2 Fine-tuning

We performed a two-step fine-tuning process. First, we fine-tuned mBART50 on the domain-specific data that was obtained using data augmentation (§3.1). Then we performed a second step of fine-tuning using the validation sets provided by the shared task organization.

2.3 Nearest Neighbor Machine Translation

To further adapt mBART50, we use the nearest neighbor machine translation approach, kNN-MT, introduced by Khandelwal et al. (2021). kNN-MT consists of a semi-parametric model: besides having a parametric component (base model) that outputs a probability distribution over the vocabulary, $p_{\mathrm{NMT}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})$, it also has a nearest neighbor retrieval mechanism, which allows direct access to a datastore of examples.

More specifically, we build a datastore $\mathcal{D}$ which consists of a key-value memory, where each entry key is the decoder's output representation, $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}_{<t}) \in \mathbb{R}^d$, and the value is the corresponding target token y_t:

$$\mathcal{D} = \{(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}_{<t}), y_t) \; \forall \, t \mid (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{S}\}, \quad (1)$$

where $\mathcal{S}$ denotes a set of parallel sentences.

Then, at inference time, the model searches the datastore to retrieve the set of k nearest neighbors $\mathcal{N}$. Using their distances $d(\cdot)$ to the current decoder's output representation, we can compute the retrieval distribution $p_{k\mathrm{NN}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})$ by applying the softmax function:

$$p_{k\mathrm{NN}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}) = \quad (2)$$
$$\frac{\sum_{(\boldsymbol{k}_j, v_j) \in \mathcal{N}} \mathbb{1}_{y_t = v_j} \exp\left(-d\left(\boldsymbol{k}_j, \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}_{<t})\right)/T\right)}{\sum_{(\boldsymbol{k}_j, v_j) \in \mathcal{N}} \exp\left(-d\left(\boldsymbol{k}_j, \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}_{<t})\right)/T\right)},$$

where T is the softmax temperature, k_j denotes the key of the j^{th} neighbor and v_j its value. Finally, the two probability distributions, $p_{\mathrm{NMT}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})$ and $p_{k\mathrm{NN}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})$, are combined to obtain the final distribution, which is used to generate the translation through beam search, by performing interpolation:

$$\begin{aligned} p(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}) = {} & (1-\lambda)\, p_{\mathrm{NMT}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}) \quad (3) \\ & + \lambda\, p_{k\mathrm{NN}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}), \end{aligned}$$

where $\lambda \in [0, 1]$ is a hyper-parameter that controls the weights given to the two distributions.

2.3.1 Using Two Datastores

As we use data from multiple sources (described in Section 3), we adapt kNN-MT to perform retrieval from two datastores which are composed of examples from different datasets. To do this, we simply need to perform retrieval from the two datastores obtaining two retrieval distributions, $p_{k\mathrm{NN}1}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})$ and $p_{k\mathrm{NN}2}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})$, computed using Eq. 2.

Then, we need to modify the distributions interpolation (Eq. 3) to account for three distributions:

$$\begin{aligned} p(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}) = {} & (1-\lambda_1-\lambda_2)\, p_{\mathrm{NMT}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}) \quad (4) \\ & + \lambda_1\, p_{k\mathrm{NN}1}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}) \\ & + \lambda_2\, p_{k\mathrm{NN}2}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}), \end{aligned}$$

where $\lambda_1 \in [0, 1]$ and $\lambda_2 \in [0, 1]$ are hyper-parameters that control the weights given to the three distributions.

3 Data

The data provided by the shared task organization is part of a corpus called MAIA corpus. It consists of parallel data of chats between an agent (English) and a customer (Brazilian Portuguese, German or French) across one domain: customer support conversation. Thus, there are a total of 6 translation directions. One of the main obstacles of this domain is the lack of parallel data publicly available. To make the task closer to a real case scenario, the shared task organization has only provided bilingual validation sets and monolingual data, for all languages.

As already mentioned, finding parallel data for this specific domain is challenging. The only exception is the dataset from WMT 2020 Shared Task on Chat Translation (Farajian et al., 2020). Unfortunately, it only contains two translation directions:

Language Direction	Original dev set	New training set	New dev set
en-de	1006	528	478
en-fr	1750	894	856
en-pt_br	1353	668	685
de-en	1103	519	584
fr-en	1003	466	537
pt_br-en	1006	469	537

Table 1: Statistics (number of sentences) of the development sets provided by the shared task organization, and of the new development and training sets after splitting it in two.

Language Direction	Number of Sentences
de-en	203,169,413
fr-en	471,885,306
pt_br-en	192,874,694

Table 2: Statistics (number of sentences) of the public available data in OPUS.

Language Direction	Number of Sentences
en-de	6494
en-fr	3311
en-pt_br	2010
de-en	6874
fr-en	1929
pt_br-en	1657

Table 3: Statistics (number of sentences) of the back-translated data.

English to German and German to English. Therefore, to circumvent the lack of domain-specific data available to fine-tune the model and to add to the datastores, we perform data augmentation (§3.1) and back-translate the monolingual data provided (§3.2).

3.1 Data Augmentation

As the domain-specific data available is limited to the provided bilingual development sets and monolingual sets, we perform data augmentation to create training sets. To do so, we use LaBSE (Language-Agnostic BERT Sentence Embedding) (Feng et al., 2020) multilingual sentence representations. In order to perform data augmentation we also use the k-nearest neighbours (kNN) implementation of the FAISS toolkit (Johnson et al., 2019).

We start by defining a seed corpus (which in our case is the validation set) and a pool corpus (generic data). Then, we use LaBSE to compute the sentence embeddings. After having the sentence embeddings, we built an in-house kNN implementation that relies on FAISS to compute the similarity among all sentences, obtaining a score between 0 (no similarity) and 1 (maximum similarity). Then, we keep the sentence-pairs with a score higher than 0.7.

3.1.1 Data Selection

Regarding data selection, we use all possible datasets publicly available in OPUS (Tiedemann, 2012) to create our pool of public data. Statistics can be find in Table 2.

3.1.2 Data Cleaning

After having downloaded all data, we perform data cleaning. To do so, we used a combination of heuristic filters and Bicleaner (Sánchez-Cartagena et al.; Ramírez-Sánchez et al., 2020). Bicleaner is a tool that detects noisy sentence-pair in a parallel corpus. It outputs the likelihood of two sentences being a mutual translation (in this case the value is near 1) or not (the value is near 0). We could have trained our own Bicleaner models but we decided to use the available ready-to-use language packages.

3.2 Back-translation

To increase the amount of domain-specific data, we also use the monolingual data provided by the shared task organizers. To do so, we performed back-translation of these datasets with best fine-tuned model using beam-search with 5 beams. The statistics are reported in Table 3. To perform back-translation we used ours models fine-tuned. We use the back-translated examples both for fine-tuning our models and as part of the datastores.

4 Experiments

In this section, we describe the experiments we made, to allow us to choose the best model to submit to the shared task.

4.1 Experimental Settings

The shared task organization provided two different baselines: one leveraging the conversation context

Model	Language Direction					
	en-de		en-fr		en-pt_br	
	SacreBLEU	COMET	SacreBLEU	COMET	SacreBLEU	COMET
Baseline (without context)	35.11	0.3989	54.23	0.8011	50.35	0.7897
Baseline (with context)	33.75	0.3755	53.95	0.8013	51.02	0.8721
kNN-MT	52.20	0.5873	61.20	0.9032	48.80	0.9398
Fine-tuned Model	62.50	0.7289	71.60	1.0485	67.80	1.1285
Fine-tuned Model + kNN-MT (1 datastore)	**62.70**	**0.7351**	71.60	1.0324	68.10	1.1330
Fine-tuned Model + kNN-MT (2 datastores)	61.30	0.7334	**72.00**	**1.0495**	**68.10**	**1.1356**

Table 4: Results obtained for the agent direction (en -> X).

Model	Language Direction					
	de-en		fr-en		pt_br-en	
	SacreBLEU	COMET	SacreBLEU	COMET	SacreBLEU	COMET
Baseline (without context)	45.75	0.5421	47.12	0.6413	44.52	0.5887
Baseline (with context)	47.13	0.6253	48.25	0.6855	47.29	0.6475
kNN-MT	57.70	0.8617	52.70	0.8390	50.90	0.7984
Fine-tuned Model	**59.40**	0.8811	**57.70**	**0.9250**	50.10	0.8117
Fine-tuned Model + kNN-MT (1 datastore)	59.20	0.8760	57.40	0.9277	50.90	0.7984
Fine-tuned Model + kNN-MT (2 datastores)	58.70	**0.8814**	57.20	0.9226	**51.80**	**0.8009**

Table 5: Results obtained for the customer direction (X -> en).

and another one that does not. Both of them use the M2M-100 (Fan et al., 2020) large pre-trained language model, which is originally a sentence-level model. Together with the baselines, the shared task organizers provided scripts to rerun the experiments using conversational context, which we did for our small test set.

As no training data was provided by the organization, we splitted the validation set into two. We used one of them as our validation set and the other was used to fine-tune the models and to perform kNN-MT. We report the data sets statistics in Table 1. We took into consideration the fact that we are dealing with conversations, and thus, we do not split conversations, i.e., we do not perform segment filtering that might break a conversation context.

We implemented all the models by the open-sourced toolkit *fairseq* (Ott et al., 2019).

Although mBART50 supports multilingual training, we trained each language direction separately. We started by fine-tuning mBART50 with the data obtained with the data augmentation process (§3.1), the data from WMT2020 Chat Translation shared task, and the back-translated monolingual data (§3.2). Then, we continued the fine-tuning step using the the training set of the shared task.

To perform retrieval we use 2 datastores having the first datastore the data from the validation sets and the second one the data obtained with the data augmentation process (§3.1) and the back-translated monolingual data (§3.2).

Hyper-Parameter	Value
Learning Rate	0.00003
Warmup updates	16000
Label Smoothing	0.2
Optimizer	Adam
β_1, β_2	0.9, 0.98
Weight Decay	0.1
Dropout	0.1
Clip Norm	5
Batch Size	256 (tokens)
Beam Size	5
kNN-MT k	8
kNN-MT temperature	10
kNN-MT λ_1	0.1
kNN-MT λ_2	0.1

Table 6: Fairseq Hyperparameters for our experiments. The first block gives the base settings used for fine-tuning mBART50 and the second block provides the details for the kNN-MT.

The selected values for hyperparameters are stated in Table 6. To evaluate the performance of our models we used SacreBLEU (Post, 2018) and COMET (Rei et al., 2020).

4.2 Results

We tested multiple configurations for kNN-MT: using only one datastore with the validation data or using two datastores with different values for the parameters that control the weight given to each distribution (λ_1 and λ_2), changing the number of neighbours retrieved, and the softmax temperature.

The results reported in Tables 4 and 5 show that performing fine-tuning of mBART50 on domain-specific data leads to large gains for all language pairs, for the two metrics. We can also see that, despite leading to worse scores than fine-tuning, simply retrieving examples from domain-specific datastores, using kNN-MT, leads to considerable gains when comparing with the baselines. Moreover, using kNN-MT with the fine-tuned model as the base model, leads to small gains on most language pairs, for the agent direction (English→X). For the customer direction (X→English), the results are very similar to the ones obtained without retrieval. When comparing with using 1 datastore (only with the validation data), using 2 datastores leads to small improvements, which suggests that the gains led by performing retrieval are due to the data coming from the validation sets.

In terms of speed, kNN-MT model requires retrieval for every single token, leading to a low decoding speed, around 8 times slower than a model that does not perform retrieval steps according to (Martins et al., 2022). Although, it is important to take into consideration that the time the model takes to add examples to the datastores is much shorter than the time needed to fine-tune the model.

Due to the repetitive nature of dialogues in customer service conversational content, we can see that by using only a few thousand domain-specific bilingual sentence-pairs together with out-of-domain sentence-pairs (selected using the data augmentation process), we are able to improve the performance of the baselines by a large margin. By analysing these experiments' results, we selected the model that combines fine-tuning and kNN-MT (with 2 datastores) as our primary submission. For the submission, we performed fine-tuning again using the complete development sets, and also added the entire development sets to the kNN-MT datastores.

5 Conclusions

We presented the joint contribution of IST and Unbabel to the WMT 2022 Chat Translation shared task. First, we perfomed fine-tuning of a large pretrained model, mBART50. Then we perfomed kNN-MT using multiple datastores to incorporate domain-specific data at decoding time. Through experiments we show that the combination of the proposed methods is a good way of performing domain adaptation when we have few domain-specific data available.

As we are dealing with conversational content it would be interesting to incorporate context information. Unfortunately the few experiments that we have performed using context did not improve the performance of our models. As future work, one interesting line of research is how to incorporate the context information together with augmented retrieval approaches. These can be complementary to each other leading to translation quality improvements.

References

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation.

M. Amin Farajian, António V. Lopes, André F. T. Martins, Sameen Maruf, and Gholamreza Haffari. 2020. Findings of the WMT 2020 shared task on chat translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 65–75, Online. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *Proc. ICLR*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*.

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. 2022. Chunk-based nearest neighbor machine translation.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael

Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón, and Sergio Ortiz-Rojas. 2020. Bifixer and bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298, Lisboa, Portugal. European Association for Machine Translation.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Víctor M. Sánchez-Cartagena, Marta Bañón, Sergio Ortiz-Rojas, and Gema Ramírez-Sánchez. Prompsit's submission to wmt 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proc. NeurIPS*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual Translation with Extensible Multilingual Pretraining and Finetuning.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. NeurIPS*.

Investigating Effectiveness of Multi-Encoder for Conversational Neural Machine Translation

Baban Gain[1], Ramakrishna Appicharla[1], Soumya Chennabasavraj[2],
Nikesh Garera[2], Asif Ekbal[1] and Muthusamy Chelliah[2]

[1]Department of Computer Science and Engineering, Indian Institute of Technology Patna, India
{gainbaban, ramakrishnaappicharla, asif.ekbal}@gmail.com
[2]Flipkart, India
{soumya.cb, nikesh.garera, muthusamy.c}@flipkart.com

Abstract

Multilingual chatbots are the need of the hour for modern business. There is increasing demand for such systems all over the world. A multilingual chatbot can help to connect distant parts of the world together, without sharing a common language. We participated in WMT22 Chat Translation Shared Task. In this paper, we report descriptions of methodologies used for participation. We submit outputs from multi-encoder based transformer model, where one encoder is for context and another for source utterance. We consider one previous utterance as context. We obtain COMET scores of 0.768 and 0.907 on English-to-German and German-to-English directions, respectively. We submitted outputs without using context at all, which generated worse results in English-to-German direction. While for German-to-English, the model achieved a lower COMET score but slightly higher chrF and BLEU scores. Further, to understand the effectiveness of the context encoder, we submitted a run after removing the context encoder during testing and we obtain similar results.

1 Introduction

Translation of Dialogues is a crucial part of building multilingual chatbots. With easier access to the internet than ever, we have the opportunity to connect with different people with different languages. However, language remains a barrier to smooth communication. Using automated machine translation systems can alleviate such issues. However, most of the general MT systems are not very suitable for conversations. This is due to additional challenges chat translation possesses that general domains do not have. This includes the presence of noisy utterances. Compared to other domains, chat is more prone to contain noisy sentences. This comes from multiple sources, as follows. a) Keyboard typos: Spelling mistakes that occurred due to quick typing. In this case, often, some characters are replaced by nearby characters on the keyboard. Further, the insertion of extra characters or the absence of some characters is also common. b) Intentional shortening of Words: Users often use short forms of words by removing certain characters (primarily vowels) while keeping the pronunciation similar to the correct word (For example, ‘hw’ instead of ‘how’). c) Grammatical Errors: Conversations usually occur in an informal setting, and grammar is mostly ignored as long as the meaning is understood correctly. However, this makes it difficult to translate. Further, there are other challenges, like context dependency. That is, the utterances can be ambiguous, and the correct meaning of an utterance can not be understood without referring to its dialogue history.

In this paper, we use a multi-encoder transformer to translate chat utterances. We use six encoder layers for source text and one encoder layer for context. For better comparison, we have submitted translations from two other models. To test the effectiveness of context, we did not provide context during the testing phase as described in section 3.3.2. Further, we train another model without using any context at all as described in 3.3.3. We achieved very competitive results for the Agent subset (English-to-German), where we obtained 0.551 BLEU, 0.730 chrF, and 0.768 COMET scores, where the best result among primary submissions of the participants are 0.555, 0.735, and 0.810 BLEU, chrF and COMET score respectively. For German-to-English, our method produced 0.907, 0.729, and 0.587 COMET, chrF, and BLEU scores, respectively.

2 Related Work

The area of chat translation mostly remained unexplored until recent years. This is in part due to the unavailability of suitable dialogue datasets. Farajian et al. (2020) introduced an German–English parallel conversational corpus. Berard et al. (2020) proposed a method that replaced rare characters

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 949 - 954
December 7-8, 2022 ©2022 Association for Computational Linguistics

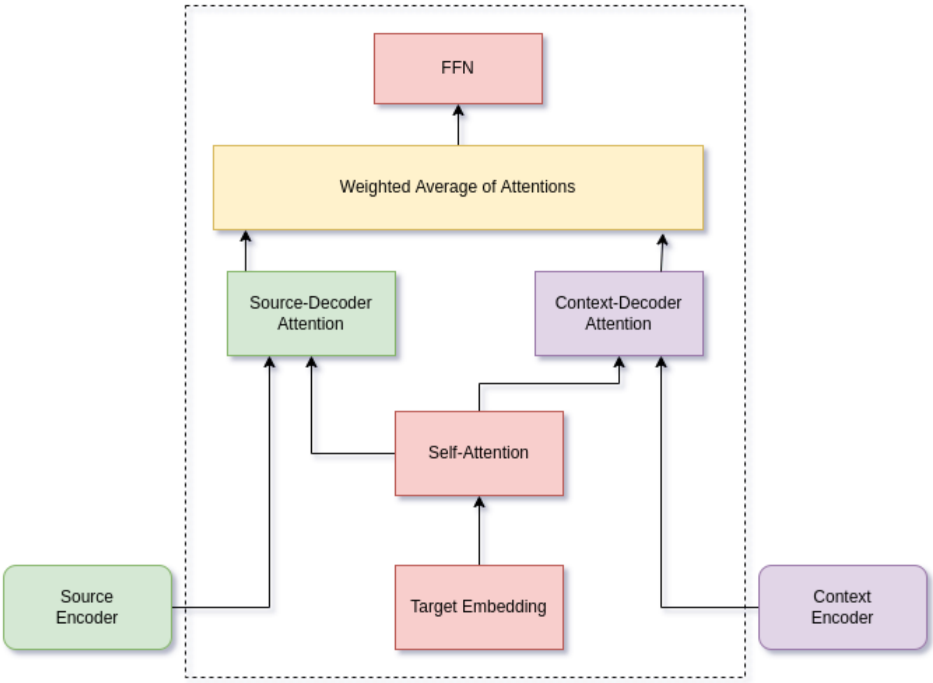


Figure 1: Diagram of our model; The weight is determined by a FFN from concatenated represenations of the attentions

with a special '<copy>' token, which helps the model to learn when to copy the tokens from source to target. Further, they used methods like inline casing, tagged back-translation (BT) (Caswell et al., 2019), Byte-Pair-Encoding (BPE) (Sennrich et al., 2016), and ensemble of models using domain-specific adaptive layers, etc. Ensemble model with a domain-specific adaptor layer generated the best translation on WMT20 Chat data. Moghe et al. (2020) used fine-tuned pre-trained models (Ng et al., 2019) on the pseudo-in-domain and in-domain data. Wang et al. (2020) used using three previous contexts along with the current sentence for adaptation of Cross-lingual Language Model Pre-training (Conneau and Lample, 2019) objectives into document-level NMT. Bao et al. (2020) used an additional encoder to process one previous context. However, adding an additional encoder did not result in consistent improvement in translation. Gain et al. (2021c) proposed a rule-based context selection technique where previous sentences by the same user are used to enhance the translation quality. This mainly helped to translate anaphoric pronouns correctly. Liang et al. (2021a) introduced a conditional variational auto-encoder (CVAE) model that captures role preference, dialogue coherence, and translation consistency. Liang et al. (2021b) proposed a multi-tasking system performing monolingual response generation, cross-lingual response generation, subsequent utterance discrimination, and speaker identification along with NMT objective. Here, the context-aware multi-tasking methods could generate better translation than context-agnostic models. Liang et al. (2022b) extended the same by introducing an additional objective, cross-lingual subsequent utterance discrimination. Further, they propose a multi-tasking algorithm that helped to generate better translation than traditional multi-tasking. Wang et al. (2021) proposed a multi-task learning-based model that identifies missing pronouns, typos and utilizes context to translate chat utterances. Liang et al. (2022a) observed visual features helps to generate better quality translation on multi-modal dialogue. Apart from chat translation, context is commonly used in other translation tasks as well. This include document translation (Kim et al., 2019; Zhang et al., 2018; Läubli et al., 2018) where other sentences from the document is used as context, multimodal translation (Yao and Wan, 2020; Gain et al., 2021a,b) where image features are used as context, etc. Gain et al. (2022) proposed a method where context is concatenated with source on both source and target side, requiring the model to translate context also, thus avoiding ignorance of context in Question-Answer translation.

3 Methodology

3.1 Pre-Training

Pre-training models with general domain data and transferring the knowledge to intended domain is standard practice in MT. We use Facebook AI's pre-trained models (Ng et al., 2019) from WMT19 [1]. The pre-training methodology consists of data processing techniques like normalize punctuation and tokenizing all data with the Moses tokenizer (Koehn et al., 2007) and byte-pair-encoding (Sennrich et al., 2016). Further, sentences with wrong language on either source or target side filtered out with language identification (Lui and Baldwin, 2012) filtering.

3.2 Model

We use a dual enocder-based transformer model. The components of the models are as follows:

- **Source Encoder:** Source Encoder consists of 6 standard transformer encoder layers. For all our models, the encoder weights are initialized from the pre-trained models. The input language of source encoder is the input language of the translation direction. That is, for English-to-German model, the language for Source Encoder is English.

- **Context Encoder:** Context Encoder of consists of 1 encoder layer. This is in part to keep model parameters lower. Further, context is supposed to assist the translation process. Thus has limited contribution compared to source. The language of the context encoder can be English or German, depending upon speaker of the previous utterance, irrespective of translation direction. We take one previous utterance from source side of previous speaker. That is, English if the speaker of previous utterance is *agent* or German if speaker of the previous utterance is *Customer*. For first utterance in a conversation, the context is empty.

- **Decoder:** Decoder consists of 6 layers of standard transformer decoder layers. We initialize the decoder from the pre-trained model. Further, in addition to encoder-decoder attention, we perform context-decoder attention. Then, we concatenate them before passing it to a feed-forward Neural Network (FFN) which determines weighted average factor g. Inspired from (Libovický et al., 2018), we take final attention output as g * context-decoder attention + (1-g) * encoder-decoder attention. The rest parts of the decoder is similar to standard transformer decoder.

3.2.1 Stage-1 Fine-tuning

For all our submissions, we perform two-stage fine-tuning. Due to the unavailability of the training set in the task, we fine-tune the model on WMT20 Chat Task (Farajian et al., 2020) data. However, since our objective is to get the highest results for WMT22 version of chat data, we use that as a validation set.

3.2.2 Stage-2 Fine-tuning

We finetune the models obtained from Stage-1 fine-tuning with WMT22 Chat Task Dev Subset. We fine-tune the models for 15 epochs. Since we are using validation set for training, we did not use any validation at this stage. We use last checkpoint from this stage as the final model and use it for testing.

3.3 Submitted Models

We submit our results for English-to-German and German-to-English directions. For each direction, we submit three results. We do not freeze any parameters during fine-tuning process for all of our submissions.

3.3.1 Primary

In our primary submission, we use the model as described in Section 3.2. We use one previous utterance as context during training, validation, and testing. This model consists of about 359M parameters.

3.3.2 Contrastive-1

Li et al. (2020) suggested that improvement in translation quality is observed after introduction of context encoder. However, it can be attributed to the contextual information acting as noise, rather than rich information relevant to the source or target. They showed that, even if context is not used during testing, the models produce similar results due to the fact that the context used during training helped the model for robust training. While this observation was for document translation, we use this method for chat translation. Thus, in this

[1] https://github.com/facebookresearch/fairseq/blob/main/examples/wmt19/README.md

Models	En-De (agent)			De-En (customer)		
	COMET	chrF	BLEU	COMET	chrF	BLEU
Baselines						
Baseline without context	0.403	0.550	0.325	0.588	0.621	0.472
Baseline with context (N=2)	0.376	0.537	0.308	0.680	0.642	0.493
Primary submissions						
BJTU-WeChat	0.810	0.735	0.555	0.946	0.775	0.649
Unbabel-IST	0.774	0.733	0.555	0.915	0.737	0.612
Our Submission	0.768	0.730	0.551	0.907	0.729	0.587
HW-TSC	0.704	0.725	0.552	0.918	0.766	0.642
Contrastive submissions						
BJTU-WeChat, C1	0.804	0.731	0.550	0.948	0.780	0.650
BJTU-WeChat, C2	0.805	0.738	0.560	0.951	0.778	0.652
Unbabel-IST, C1	0.780	0.737	0.558	0.924	0.741	0.616
Unbabel-IST, C2	0.778	0.734	0.554	0.925	0.743	0.615
Our Submission (C1)	0.769	0.730	0.551	0.905	0.729	0.587
Our Submission (C2)	0.765	0.729	0.545	0.902	0.731	0.592
HW-TSC, C1	0.649	0.670	0.473	0.909	0.755	0.618
HW-TSC, C2	0.726	0.732	0.559	0.929	0.767	0.641

Table 1: Results of submissions at WMT22 Chat task for En–De; C1: contrastive-1 submission; C2: contrastive-2 submission

submission, we use the same model as on Primary submission, but we ignore the context during testing.

	Context Encoder		Parameters	
Submission	Training	Testing	Training	Testing
Primary	Yes	Yes	359M	359M
contrastive-1	Yes	No	359M	313M
contrastive-2	No	No	313M	313M

Table 2: Comparison of methodologies for our submissions

3.3.3 Contrastive-2

We submit the results from a model without using any context for better comparison. Note that this model is trained with all other methodologies similar to Primary and Contrastive-1, which includes two-stage pre-training with the same data.

3.4 Post-Processing

We remove <unk> from the output. Further, we observe tags and modify them to the original tag, if mistranslated. For Example, we change "# PRS _ ORG #" to "#PRS_ORG#", "# Address #" to "#ADDRESS#", etc.

4 Results

We obtain a COMET (Rei et al., 2020) score of 0.768 and 0.907 on En-De and De-En directions. Further, we obtain chrF (Popović, 2015) scores of 0.730 and 0.729 for En-De and De-En. We obtain BLEU scores of 0.551 and 0.587 for Agent and Customer subsets. With contrastive-1 submission, we obtain similar results. For Agent subset, COMET score improved by 0.001 whereas, decreased by 0.002 for Customer subset. Similarly for contrastive-2 submission, COMET decreased by 0.003 whereas chrF and BLEU score decreased by 0.001 and 0.006 respectively for Agent subset. Without context method generated better results for Customer subset, improving BLEU and chrF by 0.005 and 0.002 respectively, whereas we observe a decrease of 0.005 on COMET metric. Thus, our experiment suggests that the usage of context played very limited role in the submitted systems. We suggest this is due to a lower Context Window in our experimental setting. We use only one previous sentence as a context. While it has been observed that using one context is usually sufficient on conversational or document-level datasets, WMT22 Chat Task data contain very shorter and repetitive sentences. This includes one or two word utterances (Thanks, #EMAIL#, #NAME#, Good Bye, etc), App navigational information (Tap Settings, Tap Device information, etc), etc. These utterances has very limited information to be useful as a context. Further, appearance of duplicate utterances is a challenge during training process. However, unlike general MT, conversational datasets can not be de-duplicated easily. This is because removal of some utterance from a conversation will break its structure and might not be as meaningful.

5 Conclusion

Task translation is a challenging and important task for our society. One of the major challenges in chat translation is context-dependency. We participated in WMT22 Chat Translation Task, where we submit results obtained from multi-encoder based transformer model. We obtain COMET scores of 0.768 and 0.907 on English-to-German and German-to-English directions, respectively. We found that role of context in our experimental setting is limited. In future, we would like to explore these methods with larger window size. Further, we would like to explore data de-duplication strategies for conversations.

References

Calvin Bao, Yow-Ting Shiue, Chujun Song, Jie Li, and Marine Carpuat. 2020. The University of Maryland's Submissions to the WMT20 Chat Translation Task: Searching for More Data to Adapt Discourse-Aware Neural Machine Translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 456–461.

Alexandre Berard, Ioan Calapodescu, Vassilina Nikoulina, and Jerin Philip. 2020. Naver Labs Europe's Participation in the Robustness, Chat, and Biomedical Tasks at WMT 2020. In *Proceedings of the Fifth Conference on Machine Translation*, pages 460–470.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged Back-Translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32, pages 7059–7069.

M. Amin Farajian, António V. Lopes, André F. T. Martins, Sameen Maruf, and Gholamreza Haffari. 2020. Findings of the WMT 2020 shared task on chat translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 65–75, Online. Association for Computational Linguistics.

Baban Gain, Ramakrishna Appicharla, Soumya Chennabasavraj, Nikesh Garera, Asif Ekbal, and Muthusamy Chelliah. 2022. Low resource chat translation: A benchmark for Hindi–English language pair. In *Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 83–96, Orlando, USA. Association for Machine Translation in the Americas.

Baban Gain, Dibyanayan Bandyopadhyay, and Asif Ekbal. 2021a. Experiences of adapting multimodal machine translation techniques for Hindi. In *Proceedings of the First Workshop on Multimodal Machine Translation for Low Resource Languages (MMTLRL 2021)*, pages 40–44, Online (Virtual Mode). INCOMA Ltd.

Baban Gain, Dibyanayan Bandyopadhyay, and Asif Ekbal. 2021b. IITP at WAT 2021: System description for English-Hindi multimodal translation task. In *Proceedings of the 8th Workshop on Asian Translation (WAT2021)*, pages 161–165, Online. Association for Computational Linguistics.

Baban Gain, Rejwanul Haque, and Asif Ekbal. 2021c. Not all contexts are important: The impact of effective context in conversational neural machine translation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Yunsu Kim, Duc Thanh Tran, and Hermann Ney. 2019. When and why is document-level context useful in neural machine translation? In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 24–34, Hong Kong, China. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796, Brussels, Belgium. Association for Computational Linguistics.

Bei Li, Hui Liu, Ziyang Wang, Yufan Jiang, Tong Xiao, Jingbo Zhu, Tongran Liu, and Changliang Li. 2020.

Does multi-encoder help? a case study on context-aware neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3512–3518, Online. Association for Computational Linguistics.

Yunlong Liang, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. 2021a. Modeling bilingual conversational characteristics for neural chat translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5711–5724, Online. Association for Computational Linguistics.

Yunlong Liang, Fandong Meng, Jinan Xu, Yufeng Chen, and Jie Zhou. 2022a. MSCTD: A multimodal sentiment chat translation dataset. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2601–2613, Dublin, Ireland. Association for Computational Linguistics.

Yunlong Liang, Fandong Meng, Jinan Xu, Yufeng Chen, and Jie Zhou. 2022b. Scheduled multi-task learning for neural chat translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4375–4388, Dublin, Ireland. Association for Computational Linguistics.

Yunlong Liang, Chulun Zhou, Fandong Meng, Jinan Xu, Yufeng Chen, Jinsong Su, and Jie Zhou. 2021b. Towards making the most of dialogue characteristics for neural chat translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 67–79, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jindřich Libovický, Jindřich Helcl, and David Mareček. 2018. Input combination strategies for multi-source transformer decoder. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 253–260, Brussels, Belgium. Association for Computational Linguistics.

Marco Lui and Timothy Baldwin. 2012. langid.py: An Off-the-shelf Language Identification Tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea.

Nikita Moghe, Christian Hardmeier, and Rachel Bawden. 2020. The University of Edinburgh-Uppsala University's Submission to the WMT 2020 Chat Translation Task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 471–476.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 News Translation Task Submission. In *Proceedings of the Fourth Conference on Machine Translation*, pages 314–319, Florence, Italy.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Longyue Wang, Zhaopeng Tu, Xing Wang, Li Ding, Liang Ding, and Shuming Shi. 2020. Tencent AI Lab Machine Translation Systems for WMT20 Chat Translation Task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 483–491.

Tao Wang, Chengqi Zhao, Mingxuan Wang, Lei Li, and Deyi Xiong. 2021. Autocorrect in the process of translation — multi-task learning improves dialogue machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 105–112, Online. Association for Computational Linguistics.

Shaowei Yao and Xiaojun Wan. 2020. Multimodal transformer for multimodal machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4346–4350, Online. Association for Computational Linguistics.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542, Brussels, Belgium. Association for Computational Linguistics.

BJTU-WeChat's Systems for the WMT22 Chat Translation Task

Yunlong Liang[1]*, Fandong Meng[2], Jinan Xu[1]†, Yufeng Chen[1] and Jie Zhou[2]
[1]Beijing Key Lab of Traffic Data Analysis and Mining,
Beijing Jiaotong University, Beijing, China
[2]Pattern Recognition Center, WeChat AI, Tencent Inc, China
{yunlongliang,jaxu,chenyf}@bjtu.edu.cn
{fandongmeng,withtomzhou}@tencent.com

Abstract

This paper introduces the joint submission of the Beijing Jiaotong University and WeChat AI to the WMT'22 chat translation task for English⇔German. Based on the Transformer (Vaswani et al., 2017), we apply several effective variants. In our experiments, we utilize the pre-training-then-fine-tuning paradigm. In the first pre-training stage, we employ data filtering and synthetic data generation (i.e., back-translation, forward-translation, and knowledge distillation). In the second fine-tuning stage, we investigate speaker-aware in-domain data generation, speaker adaptation, prompt-based context modeling, target denoising fine-tuning (Meng et al., 2020), and boosted self-COMET-based model ensemble. Our systems achieve 0.810 and 0.946 COMET (Rei et al., 2020) scores[1] on English→German and German→English, respectively. The COMET scores of English→German and German→English are the highest among all submissions.

1 Introduction

We participate in the WMT 2022 shared task on chat translation in two language directions, English→German and German→English. In this year's chat translation task, we apply the two-stage training strategy. In the first stage, we investigate model architecture and data augmentation. In the second stage, we mainly focus on exploiting speaker-aware in-domain data augmentation, speaker adaptation, prompt-based context modeling, target denoising fine-tuning (Meng et al., 2020), and model ensemble strategies. This task aims to build machine translation systems to translate conversational text and thus supports fluent communication between an agent speaking in English and a customer speaking in a different language (e.g., German), which is different from the first pre-training stage (Farajian et al., 2020; Liang et al., 2021a, 2022a; Liu et al., 2021; Gain et al., 2021, 2022; Buschbeck et al., 2022). Therefore, we mainly pay attention to the second fine-tuning stage.

In the first pre-training stage, we follow previous work (Meng et al., 2020; Zeng et al., 2021; Meng and Zhang, 2019; Yan et al., 2020) and utilize several effective Transformer variants. Specifically, we combine the Multi-Head-Attention (Vaswani et al., 2017), Average Attention Transformer (Zhang et al., 2018), and Talking-Heads Attention (Shazeer et al., 2020), which have shown significant model performance and diversity. For data augmentation, we employ the back-translation method to use the target-side monolingual data and apply the forward-translation to leverage the source-side monolingual data. To fully utilize the source-side of bilingual data, we use the sequence-level knowledge distillation method (Kim and Rush, 2016).

In the second fine-tuning stage, for speaker-aware in-domain data augmentation, based on the BConTrasT (Farajian et al., 2020) dataset of the WMT20 chat translation task, we firstly adapt our pre-trained model to each speaker by using the speaker tag as a pseudo token and then apply it to the Taskmaster-1 (Byrne et al., 2019) corpus to generate the speaker-aware in-domain data. For speaker adaptation, we follow previous work (Moghe et al., 2020) to prepend the corresponding speaker tag to each utterance on both the source and the target side to get a speaker-aware dataset. For prompt-based context modeling, we exploit the prompt learning to incorporate the bilingual context and then apply the target denoising fine-tuning method (Meng et al., 2020) to train our model. For the model ensemble, inspired by Zeng et al. (2021), we select high-potential can-

[*] Work was done when Yunlong was interning at Pattern Recognition Center, WeChat AI, Tencent Inc, China.

[†] Jinan Xu is the corresponding author.

[1] The COMET is the official automatic evaluation metric.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 955 - 961
December 7-8, 2022 ©2022 Association for Computational Linguistics

didate models from two aspects, namely model performance (COMET scores) and model diversity (Self-COMET scores among all candidate models). Based on this, we design a search algorithm to gradually select the current best model of the model candidate pool for the final model ensemble.

2 Model Architectures

In this section, we describe the model architectures we used in two translation directions, where we mainly follow the previous state-of-the-art models (Zeng et al., 2021). We also refer readers to read the paper for details.

2.1 Model Configurations

Given the strong capacity of deeper and wider architectures, we use them in our experiments. Specifically, following Zeng et al. (2021), we use 20-layer encoders for deeper models and set the hidden size to 1024 for all models. We set the decoder depth to 10. For the wider ones, we adopt 12 encoder layers, 2048 for hidden size, and 8192 to 15000 for filter sizes.

2.2 Transformer Variants

Average Attention Transformer. Following Zeng et al. (2021), the average attention transformer (Zhang et al., 2018) are employed to add model diversity. In the AAN, the context representation g_i for each input embedding is calculated as follows:

$$g_i = \mathrm{FFN}(\frac{1}{i}\sum_{k=1}^{t} y_k),$$

where y_k is the input embedding for step k and t is the current time step. FFN denotes the position-wise feed-forward network (Vaswani et al., 2017).

Talking Heads Attention. Similarly, talking-heads attention (Shazeer et al., 2020) also performs well in Zeng et al. (2021), which can transform the attention-logits and the attention scores and thus allow information interaction among attention heads by adding two linear projection layers W_l and W_a:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{k}}W_l)W_aV.$$

3 System Overview

In this section, we describe our system used in the WMT 2022 chat translation shared task, which includes two parts, namely general pre-training and in-domain fine-tuning. The pre-training part includes data filtering and synthetic data generation. The in-domain fine-tuning consists of speaker-aware in-domain data generation, speaker adaptation, prompt-based context modeling, the target denoising fine-tuning (Meng et al., 2020), and boosted Self-COMET-based model ensemble.

3.1 General Pre-training

3.1.1 Data Filtering

We filter the bilingual training corpus (including synthetic parallel data) with the following rules (Zeng et al., 2021): 1) Normalize punctuation; 2) Remove the sentence whose length is more than 100 words or a single word that exceeds 40 characters; 3) Filter out the duplicated sentence pairs; 4) Delete the sentence whose word ratio between the source and the target words exceeds 1:4 or 4:1.

3.1.2 Synthetic Data Generation

For data augmentation, we obtain the general domain synthetic data via back-translation, forward-translation, and knowledge distillation.

Tagged Back-Translation. Previous work has shown that different methods of generating pseudo corpus have a different influence on translation performance (Edunov et al., 2018; Hoang et al., 2018; Zeng et al., 2021). Following them, we attempt two generating strategies: 1) Beam Search: produce translation by beam search (beam size = 5). 2) Sampling Top-k: Select a word randomly from top-k (k = 15) words when inference.

Forward-Translation. We then ensemble models to forward-translate the monolingual data of the source language to further enhance model performance. We obtain a stable improvement in both directions, which is consistent with previous work (Zeng et al., 2021).

Knowledge Distillation. Knowledge Distillation aims to transfer knowledge from the teacher model to student models, which has shown effective for NMT (Kim and Rush, 2016; Wang et al., 2021; Zeng et al., 2021). Specifically, we first use the teacher model to generate synthetic corpus in the forward direction (i.e., En→De). Then, we train our student models with the generated corpus.

Note that we prefix all the synthetic sentences by appending a pseudo tag <BT> when jointly training with genuine data.

3.2 In-domain Fine-tuning

3.2.1 Speaker-aware In-domain Data Generation

Inspired by Moghe et al. (2020), we prepend the corresponding speaker tag (the `<agent>` or the `<customer>`) to each utterance on both the source and the target side to get a speaker-aware dataset based on the BConTrasT dataset of the WMT20 chat translation task (Farajian et al., 2020). Secondly, we adapt our pre-trained model to each speaker on the speaker-aware dataset. Then, we apply the adapted model to the monolingual Taskmaster-1 (Byrne et al., 2019) corpus, which is the original source of BConTrasT (Farajian et al., 2020), to generate the speaker-aware in-domain data.

3.2.2 Speaker Adaptation

As a special characteristic of chat translation, distinguishing between the two speaker roles plays an important role as they both form the complete dialogue. And modeling the speaker characteristic has been demonstrated effective in previous work (Moghe et al., 2020; Liang et al., 2021c, 2022b, 2021b, 2022c). Therefore, our data used in the fine-tuning has a corresponding speaker tag (the `<agent>` or the `<customer>`) appended in the first token of each utterance.

3.2.3 Prompt-based Context Modeling

Previous studies (Wang et al., 2020; Moghe et al., 2020) have shown that the multi-encoder framework cannot improve the model performance after using the context in the chat translation task, while a unified model (Ma et al., 2020; Liang et al., 2021c) can. Therefore, we also investigate incorporating the context in the unified model with prompt learning (without modifying the model architecture). Specifically, we add two preceding bilingual contexts at the tail of each utterance with an indicator `<context begins>`, where we also use a special tag `<SEP>` to separate different utterances of the bilingual context. In this way, our model with context modeling can achieve a better COMET.

3.2.4 Target Denoising Fine-tuning

To bridge the exposure bias (Ranzato et al., 2016), we add noisy perturbations into decoder inputs when fine-tuning. Therefore, the model becomes more robust to prediction errors by target denoising fine-tuning (Zhang et al., 2019; Meng et al., 2020). Specifically, the fine-tuning data generator chooses 30% of utterance pairs (Note that we do not include the indicator word and the bilingual context) to add noise and keeps the remaining 70% of sentence pairs unchanged. For a chosen pair, we keep the source sentence untouched and replace the i-th token of the target sentence with (I) a random token of the current target sentence in 15% probability and (II) the unchanged i-th in 85% probability.

Algorithm 1: Boosted Self-COMET-based Ensemble (BSCE)

```
Input:
    List of candidate models $\mathbb{M}$ = {$m_i$, ..., $m_n$}
    Valid set COMET for each model $\mathbb{C}$ = {$c_i$, ..., $c_n$}
    Average Self-COMET for each model $\mathbb{S}$ = {$s_i$, ..., $s_n$}
    The number of models $n$
    The number of ensemble models $e$
Output: Selected Model Pool $\mathbb{P}$
for $i \leftarrow 1$ $to$ $n$ do
    $weight = \frac{(max(\mathbb{S})-min(\mathbb{S}))}{(max(\mathbb{C})-min(\mathbb{C}))}$
    $score_i =$
        $(c_i - min(\mathbb{C})) \cdot weight + (max(\mathbb{S}) - s_i)$
end for
Add the highest score model to candidates list
    $\mathbb{P}$ = { $m_{top}$ }
while $|\mathbb{P}| < e$ do
    $index = \arg\min_i \frac{1}{|\mathbb{M}-\mathbb{P}|} \sum_{i \in \mathbb{M}-\mathbb{P}, j \in \mathbb{P}} BLEU(i,j)$
    Add $m_{index}$ to candidate list $\mathbb{P}$
end while
return $\mathbb{P}$
```

3.2.5 Boosted Self-COMET-based Model Ensemble (BSCE)

After we get plenty of fine-tuned models, how to search for the best combination for the ensemble model is a difficult question. Inspired by Zeng et al. (2021), we propose a Boosted Self-COMET-based Ensemble (BSCE) algorithm, as shown in algorithm 1. Since the existing boosted Self-BLEU-based pruning strategy (Zeng et al., 2021) is designed for achieving higher BLEU scores with high efficiency, it can not help obtain better COMET scores. Therefore, we adapt it to COMET scores. Then, we can obtain the best ensemble models from n top models by a greedy search strategy.

The algorithm takes as input a list of n strong single models $\mathbb{M}$, COMET scores on the development set for each model $\mathbb{C}$, average Self-COMET scores for each model $\mathbb{S}$, the number of models n, and the expected number of ensemble models e. The algorithm returns a set $\mathbb{P}$ consisting of e selected models. We calculate the weighted score for each model (line 2). The weight (line 3) calculated is a trade-off between the development set COMET score and the Self-COMET score since the performance and the diversity play the same key role in ensemble (Zeng et al., 2021). Then the set $\mathbb{P}$ initially contains the model m_{top} has the highest weighted score. Next, we iteratively re-compute the average Self-COMET between the remaining models in '$\mathbb{M} - \mathbb{P}$' and selected models in $\mathbb{P}$, based on which we select the model that has a minimum Self-COMET score into $\mathbb{P}$.

4 Experiments and Results

4.1 Setting

The implementation of our models is based on Fairseq[2]. All the single models in the first pre-training stage are carried out on 8 NVIDIA V100 GPUs (32 GB memory of each). And all the models in the second fine-tuning stage are conducted on 4 NVIDIA V100 GPUs. We use the Adam optimizer with β_1 = 0.9, β_2 = 0.998. The batch size are set to 8192 and 4096 tokens per GPU for pre-training and fine-tuning, respectively. We set the "update-freq" parameter to 2 and 1 for both stages. The learning rate is set to 0.0005 and 0.0004 for two stages, respectively. We use the warmup step to 4000. We calculate COMET[3] score for all experiments which is officially recommended.

English and German sentences are segmented by Moses[4]. We apply punctuation normalization and Truecasing. We use byte pair encoding BPE (Sennrich et al., 2016) with 32K operations. For the post-processing, we apply de-truecaseing and detokenizing on the English and German translations with the scripts provided in Moses.

4.2 Dataset

The data statistics of the two stages are shown in Table 1. For the general pre-training, the bilingual data is the combination of all parallel data in WMT21. For monolingual data, we use the News Crawl, Common Crawl, and Extended Common Crawl. For synthetic data generation, we back-translate all the target monolingual data and forward-translate the source monolingual data. For the in-domain fine-tuning, we use all the training, valid, and testing data of the wmt20 chat task as our training data. For monolingual data, we select the Taskmaster-1 (Byrne et al., 2019) corpus to build the pseudo-paired data using the method described in Section 3.2.1.

	General pre-training	In-domain fine-tuning
Bilingual Data	74.8M	17,847
Source Mono Data	332.8M	302,079
Target Mono Data	237.9M	-

Table 1: Statistics of all training data.

Models	**En→De**	**De→En**
Chat baseline w/o context	0.403	0.588
Chat baseline w context	0.376	0.680
Pre-trained deeper model w/o context	0.544	0.865
+ in-domain genuine data w/ context (FT1)	0.772	0.905
+ in-domain pseudo data w/ context (FT2)	0.767	0.903
+ in-domain both data w/ context (FT3)	0.781	0.908
Pre-trained wider model w/o context	0.604	0.879
+ in-domain genuine data w/ context (FT4)	0.782	0.908
+ in-domain pseudo data w/ context (FT5)	0.779	0.906
+ in-domain both data w/ context (FT6)	**0.785**	**0.909**

Table 2: COMET scores on the Valid set for both pre-trained models, and each of fine-tuned on (i) in-domain genuine data, (ii) in-domain pseudo data, and (iii) both in-domain data.

4.3 Results

We report COMET scores (Rei et al., 2020) on the validation set (generally, beam size = 5 and length penalty = 0.6).

Pre-training and Fine-tuning. The results in Table 2 show that all pre-trained models outperform the baseline models trained on the chat training data. We observe that in-domain fine-tuning of the pre-trained models always gives large gains even on the in-domain pseudo data. We also find that the performance of different model architectures comes close after in-domain fine-tuning. Though these models perform similarly, as they have different architectures or are trained on different data, they generate diverse translations and show a cumulative effect when ensemble.

Final Submissions. Table 3 shows the results of our primary submission on both the validation and test set. Note that all candidate models with different architectures or trained with different data are used for the ensemble. We find that our BSCE

[2] https://github.com/pytorch/fairseq

[3] https://unbabel.github.io/COMET/html/index.html

[4] http://www.statmt.org/moses/

Models	En→De	De→En
Best Single Model	0.785	0.909
+ Normal Ensemble	0.788	0.908
+ BSCE	0.790	0.911
+ BSCE + Large beam (*)	**0.792**	**0.913**
Official results on the Test set		
+ BSCE + Large beam (*)	**0.810**	**0.946**
Best Official	**0.810**	**0.946**

Table 3: Valid set COMET scores for ensemble with different strategies and the official COMET results of our submissions. '*' indicates the primary system of our submissions.

Models	En→De	De→En
FT6 + no tag	0.779	0.904
FT6 + speaker	**0.785**	**0.909**

Table 4: Valid set COMET scores for fine-tuning with speaker tags .

is effective in both directions (more analyses are shown in Section 5.3). Inspired by Wang et al. (2020), we also tried large beam size. Finally, our primary system achieves the highest results among all submissions[5].

5 Analysis

5.1 Effect of Speaker Tags

As shown in Table 4, we observe that the performance in both directions improves with the addition of tags, which is consistent with Moghe et al. (2020). It shows that adding the speaker tag indeed can improve the chat translation performance.

5.2 Effect of Prompt-based Context Modeling (PCM)

As shown in Table 5, we investigate the effect of the context. The bilingual context involves the utterance in mixed language. Therefore, we investigate the different contexts with prompt learning. The results show that the models achieve slight performance gains with suitable context. And using context in the same language was more beneficial than the mixed context, which is consistent with previous work (Moghe et al., 2020).

5.3 Effect of Boosted Self-COMET-based Ensemble (BSCE)

Inspired by the boosted Self-BLEU-based ensemble (Zeng et al., 2021), we propose the Boosted Self-COMET-based Ensemble. To verify its superiority, we first select the top 10 models with different architecture and training data. The results are shown in the "+Normal Ensemble" of Table 3. For the BSCE, we need to get the translation result of every model to calculate the Self-COMET. After that, we only need to perform the inference process once. Then, we can select the best models for the ensemble. Here, we select 10 models and 4 models for **En→De** and **De→En**, respectively. The results are shown in "+BSCE" of Table 3. Based on it, we obtain better results after using the large beam (beam sizes of 9 and 8 for **En→De** and **De→En**, respectively). These results show the effectiveness of our BSCE method.

Models	En→De	De→En
FT6 + w/o context	**0.782**	**0.905**
using previous context (mix language)		
FT6 + w/ PCM (+ 1 prev)	0.781	**0.905**
FT6 + w/ PCM (+ 2 prev)	0.779	0.901
FT6 + w/ PCM (+ 3 prev)	0.775	0.897
using previous context (same language)		
FT6 + w/ PCM (+ 1 prev)	**0.785**	**0.909**
FT6 + w/ PCM (+ 2 prev)	0.784	**0.909**
FT6 + w/ PCM (+ 3 prev)	0.782	0.904

Table 5: Valid set COMET scores for fine-tuning with different contexts. The numbers before "prev" indicate the number of preceding utterances used as context.

6 Conclusions

We investigate the pre-training-then-fine-tuning paradigm to build chat translation systems, which are some effective transformer-based architectures. Our systems are also built on several popular data augmentation methods such as back-translation, forward-translation, and knowledge distillation. In the fine-tuning, we enhance our system by speaker-aware in-domain data generation, speaker adaptation, prompt-based context modeling, target denoising fine-tuning (Meng et al., 2020), and boosted self-COMET-based model ensemble. Our systems achieve 0.810 and 0.946 COMET (Rei et al., 2020) scores on English→German and German→English, respectively. These COMET scores are the highest among all submissions.

Acknowledgements

The research work described in this paper has been supported by the National Key R&D Program of China (2020AAA0108005) and the National Na-

[5] https://wmt-chat-task.github.io/

ture Science Foundation of China (No. 61976015, 61976016, 61876198 and 61370130). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

Bianka Buschbeck, Jennifer Mell, Miriam Exel, and Matthias Huck. 2022. “hi, how can i help you?” improving machine translation of conversational content in a business context. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 189–198.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. In *Proceedings of EMNLP-IJCNLP*, pages 4516–4525.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of EMNLP*, pages 489–500, Brussels, Belgium.

M. Amin Farajian, António V. Lopes, André F. T. Martins, Sameen Maruf, and Gholamreza Haffari. 2020. Findings of the WMT 2020 shared task on chat translation. In *Proceedings of WMT*, pages 65–75.

Baban Gain, Ramakrishna Appicharla, Asif Ekbal, Muthusamy Chelliah, Soumya Chennabasavraj, and Nikesh Garera. 2022. Low resource chat translation: A benchmark for hindi–english language pair. In *Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 83–96.

Baban Gain, Rejwanul Haque, and Asif Ekbal. 2021. Not all contexts are important: The impact of effective context in conversational neural machine translation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of WMT*, pages 18–24, Melbourne, Australia.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of EMNLP*, pages 1317–1327, Austin, Texas.

Yunlong Liang, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. 2021a. Modeling bilingual conversational characteristics for neural chat translation. In *Proceedings of ACL*, pages 5711–5724.

Yunlong Liang, Fandong Meng, Jinan Xu, Yufeng Chen, and Jie Zhou. 2022a. MSCTD: A multimodal sentiment chat translation dataset. In *Proceedings of ACL*, pages 2601–2613, Dublin, Ireland.

Yunlong Liang, Fandong Meng, Jinan Xu, Yufeng Chen, and Jie Zhou. 2022b. Scheduled multi-task learning for neural chat translation. In *Proceedings of ACL*, pages 4375–4388, Dublin, Ireland.

Yunlong Liang, Fandong Meng, Ying Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2021b. Infusing multi-source knowledge with heterogeneous graph neural network for emotional conversation generation. In *Proceedings of AAAI*, volume 35, pages 13343–13352.

Yunlong Liang, Fandong Meng, Ying Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2022c. Emotional conversation generation with heterogeneous graph neural network. *Artificial Intelligence*, 308:103714.

Yunlong Liang, Chulun Zhou, Fandong Meng, Jinan Xu, Yufeng Chen, Jinsong Su, and Jie Zhou. 2021c. Towards making the most of dialogue characteristics for neural chat translation. In *Proceedings of EMNLP*, pages 67–79.

Siyou Liu, Yuqi Sun, and Longyue Wang. 2021. Recent advances in dialogue machine translation. *Information*, 12(11):484.

Shuming Ma, Dongdong Zhang, and Ming Zhou. 2020. A simple and effective unified encoder for document-level machine translation. In *Proceedings of ACL*, pages 3505–3511.

Fandong Meng, Jianhao Yan, Yijin Liu, Yuan Gao, Xianfeng Zeng, Qinsong Zeng, Peng Li, Ming Chen, Jie Zhou, Sifan Liu, and Hao Zhou. 2020. WeChat neural machine translation systems for WMT20. In *Proceedings of WMT*, pages 239–247, Online.

Fandong Meng and Jinchao Zhang. 2019. DTMT: A novel deep transition architecture for neural machine translation. In *Proceedings of AAAI*, pages 224–231.

Nikita Moghe, Christian Hardmeier, and Rachel Bawden. 2020. The university of edinburgh-uppsala university’s submission to the wmt 2020 chat translation task. In *Proceedings of WMT*, pages 471–476.

Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *Proceedings of ICLR*.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of EMNLP*, pages 2685–2702, Online.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of ACL*, pages 1715–1725, Berlin, Germany.

Noam Shazeer, Zhenzhong Lan, Youlong Cheng, Nan Ding, and Le Hou. 2020. Talking-heads attention.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NIPS*, pages 5998–6008.

Fusheng Wang, Jianhao Yan, Fandong Meng, and Jie Zhou. 2021. Selective knowledge distillation for neural machine translation. In *Proceedings of ACL*, pages 6456–6466, Online.

Longyue Wang, Zhaopeng Tu, Xing Wang, Li Ding, Liang Ding, and Shuming Shi. 2020. Tencent ai lab machine translation systems for wmt20 chat translation task. In *Proceedings of WMT*, pages 481–489.

Jianhao Yan, Fandong Meng, and Jie Zhou. 2020. Multi-unit transformers for neural machine translation. In *Proceedings of EMNLP*, pages 1047–1059, Online.

Xianfeng Zeng, Yijin Liu, Ernan Li, Qiu Ran, Fandong Meng, Peng Li, Jinan Xu, and Jie Zhou. 2021. WeChat neural machine translation systems for WMT21. In *Proceedings of WMT*, pages 243–254, Online. Association for Computational Linguistics.

Biao Zhang, Deyi Xiong, and Jinsong Su. 2018. Accelerating neural transformer via an average attention network. In *Proceedings of ACL*, pages 1789–1798, Melbourne, Australia. Association for Computational Linguistics.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the gap between training and inference for neural machine translation. In *Proceedings of ACL*, pages 4334–4343, Florence, Italy.

HW-TSC Translation Systems for the WMT22 Chat Translation Task

Jinlong Yang, Zongyao Li, Daimeng Wei, Hengchao Shang, Xiaoyu Chen,
Zhengzhe Yu, Zhiqiang Rao, Shaojun Li, Zhanglin Wu, Yuhao Xie,
Yuanchang Luo, Ting Zhu, Yanqing Zhao, Lizhi Lei, Hao Yang, Ying Qin
Huawei Translation Service Center, Beijing, China
{yangjinlong7,lizongyao,weidaimeng,shanghengchao,chenxiaoyu35,
yuzhengzhe,raozhiqiang,lishaojun18,wuzhanglin2,xieyuhao2,luoyuanchang,
zhuting20,zhaoyanqing,leilizhi,yanghao30,qinying}@huawei.com

Abstract

This paper describes the submissions of Huawei Translation Services Center(HW-TSC) to WMT22 chat translation shared task on English↔German (en-de) bidirection with results of zero-shot and few-shot tracks. We use the deep transformer architecture with a larger parameter size. Our submissions to the WMT21 News Translation task are used as the baselines. We adopt strategies such as back translation, forward translation, domain transfer, data selection, and noisy forward translation in task, and achieve competitive results on the development set. We also test the effectiveness of document translation on chat tasks. Due to the lack of chat data, the results on the development set show that it is not as effective as sentence-level translation models.

1 Introduction

Neural Machine Translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2015; Gehring et al., 2017) has achieved good translation results in most scenarios, but few researches have been done in the field of chat translation, mainly because of insufficient chat data.

WMT20 holds the chat translation shared task (Farajian et al., 2020) for the first time. The data set mainly includes pre-sales conversations between customers and agents (meal booking, air ticket reservation, etc.). This year, the data set focuses on post-sales conversations between customers and agents. Although the translation content is all about chat, the domains are slightly different. The results show that the data from previous years can not effectively improve the quality of the model for this year's task.

We participate in the en-de bidirectional translation task. The en-de bidirectional models we submitted to the WMT21 news task (Wei et al., 2021) are used as the baseline models and the architecture is deep transformer (Vaswani et al., 2017; Dou et al., 2018). Commonly-used optimization strategies are used, such as domain transfer, data selection, back translation, self-training, noisy self-training, finetuning and model averaging.

Considering that the chat task is a context-aware translation task, we conduct a series of document-level (Wang et al., 2017) experiments using WMT document data, but it does not work well on development sets. The analysis shows that the document data deviates greatly from the chat domain, and the data therefore cannot effectively improve chat translation quality. According to the results, the best models are obtained by selecting in-domain data from out domain data by the development sets.

This paper is structured as follows: Section 2 describes our data volume and data pre-processing method. The model structure and method we used are presented in Section 3. Section 4 details our experiment setting. We present the results in Section 5, and finally we conclude our work in Section 6.

2 Data

2.1 Data Size

We use WMT21 news en-de bidirection models as our baselines (Wei et al., 2021). Bilingual data comes for WMT20 chat task Farajian et al. (2020), and monolingual data is from Byrne et al. (2019)

We select data of three related domains, including conversation, subtitle, and shopping, from our in-house English corpus for domain transfer. In addition, the document-level data from WMT22 general task [1] is used to train the document-level translation model. In addition, 40M in-house general bilingual data is used.

For details about the data size, see Table 1 and Table 2.

[1] Data is available from https://www.statmt.org/wmt22/

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 962 - 968
December 7-8, 2022 ©2022 Association for Computational Linguistics

	general	chat 20	chat 22	doc
en-de	40M	17847	2109	400K

Table 1: Sentences size of bilingual data used for training

	Domain-related	chat 20	chat 22	doc
en	5M	1M	6389	20M
de	-	-	7011	20M

Table 2: Sentences size of monolingual data used for training

2.2 Data pre-processing

Considering that the data sizes of WMT20 and WMT22 chat tasks are limited, we do not cleanse the chat data. We use the following data cleansing methods for other data:

- Remove duplicate sentences (Khayrallah and Koehn, 2018; Ott et al., 2018)
- Filter out sentences with more than 150 words
- Filter out sentences with length ratios greater than 1.5
- Apply langid (Joulin et al., 2016, 2017) to filter out sentences in other languages
- Use fast-align (Dyer et al., 2013) to filter out sentence pairs that are poorly aligned.

Besides, we adopt joint SentencePiece Model(SPM) (Kudo and Richardson, 2018; Kudo, 2018) for word segmentation with a vocabulary of 32K.

3 System Overview

3.1 Model

Transformer has been widely used for neural machine translation in recent years, which has achieved good performance even with the most primitive architecture. Therefore, the baseline models for WMT21 news en-de task use the Transformer-Big architecture. Deep transformer is an improvement of Transformer, which increases the number of encoder layers and uses pre-layer-normalization to further improve model performance. Therefore, in this task, we adopt the following model architecture:

- Deep 25-6 large Model: This model features 25-layer encoder, 6-layer decoder, 1024 dimensions of word vector, 4096 domensions of FFN, 16-head self-attention, and pre-layer-normalization.

3.2 Document-level NMT

Document-level machine translation (Ma et al., 2021) conditions on surrounding sentences to produce coherent translations. There has been a lot of work on custom model architectures to integrate document context into translation models.

There are many document translation strategies, such as Doc2Sent, Window2Window, Doc2Doc (Junczys-Dowmunt, 2019), DocBT (Junczys-Dowmunt, 2019), DocRepair (Voita et al., 2019), NoisyChannelDoc (Yu et al., 2019) and G-Transformer (Bao et al., 2021). Among the methods mentioned above, Doc2Doc and DocBT are preferred by us because the data processing procedures are simple and the model requires no modification.

To train our document-level model, the bilingual document data is spliced into a long sequence based on paragraph information and the sentences are separated by numbered <SEPX> symbols. For document-level monolinguals, we first generate synthetic bilingual data by back translation and use the same strategy to construct doc2doc data. We then use the document data to fine-tune the sentence-level translation model to ensure the model capable of translating long sequences.

We use two methods for inference. The first one translates single sentences just like a standard translation model. The other method combines a sentence with its context to construct a long-sequence input. After decoding, the model splits the result into single sentences and sacreBLEU[2] (Post, 2018) is calculated on the single sentences.

3.3 Data Selection

Data selection (van der Wees et al., 2017) is a data augmentation method that we use to select in-domain data from out-of-domain data.

For monolingual data selection, we train a FastText (Joulin et al., 2016) classifier using a small number of English monolinguals in subtitle, conversation, and shopping domains, and then select in-domain English monolinguals from the common corpus.

For bilingual data selection, as mentioned by Wang et al. (2019, 2018) , we use the in-domain data to fine-tune the out-domain model, and then

[2] https://github.com/mjpost/sacrebleu

System	20 en→de test	20 de→en test	22 en→de dev	22 de→en dev
baseline	45.1	46.7	50.3	58.7
+ Data Selection	49.2(+4.1)	48.1(+1.4)	62.5(+12.2)	65.5(+6.8)
+ Noisy FT	49.0(+3.9)	49.9(+3.2)	**64.3**(+13.1)	65.5(+6.8)
+ Model Average	49.8(+4.7)	49.2(+2.5)	63.2(+12.9)	**65.7**(+7.0)

Table 3: sacreBLEU score on chat20 test set and chat22 dev set

use the model before and after the fine-tuning to calculate the decoding probability score of the out-domain bilingual data. The data with a higher score on the fine-tuned model is selected as the in-domain bilingual data. The specific scoring is carried out according to the formula 1.

$$score = \frac{\log P(y|x;\theta_{in}) - \log P(y|x;\theta_{out})}{|y|} \quad (1)$$

Where θ_{out} represents the model trained with out-domain data, and θ_{in} represents the model after fine-tuning with a small amount of in-domain bilingual data, and |y| represents the length of the target sentence.

3.4 Forward Translation

Forward Translation (FT) (Wu et al., 2019), also known as Self-Training (Imamura and Sumita, 2018) , usually refers to using a forward NMT model to translate source-side monolingual data to target-side text so as to generate synthetic bilinguals. The data is then used to train the forward translation model. Generally, beam search (Freitag and Al-Onaizan, 2017) is used for forward translation. In our experiment, the beam size is set to 4.

Noisy self-training (He et al., 2020) adds noise to the source-side of the pseudo parallel corpus generated by forward translation. Experiments show that this method is effective in low resource tasks. Noisy self-training is therefore used in the last step when a small amount of in-domain monolinguals is used.

3.5 Back Translation

Back-translation(BT) (Edunov et al., 2018) has been recognized as the most effective data augmentation strategy for enhancing NMT model performance. Contrary to forward translation, it translates target-side monolinguals into source-side to generate synthetic parallel corpus. Among the many back translation methods, sampling (Graça et al., 2019), noise (Edunov et al., 2018) and tagged back translation (Caswell et al.) work better. In our experiment, sampling back-translation is chosen.

3.6 Fine-tuning

Fine-tuning (Dakwale and Monz, 2017) is a way to achieve domain transfer. In our translation task, we adopt a three-stage fine-tuning strategy. Firstly, we use synthetic corpus from similar domains to fine-tune the out-of-domain NMT model, and then use bilingual data selected from general domain according to the development set to improve the model performance. After that, we use the synthetic data generated from the in-domain monolingual data to fine-tune the in-domain model for more fine-grained domain transfer.

3.7 Model Averaging

Model averaging (Dormann et al., 2018) is a commonly used technique to improve translation quality. Generally, models (5 in our experiment) that perform best on the development set are selected for parameter averaging, result to significantly improvement.

4 Experiment Setting

During the training phase, we use Pytorch-based Fairseq[3] (Ott et al., 2019) open-source framework as our benchmark system. Each model is trained using 8 GPUs with a batch size of 2048. The update frequency is 4 and the learning rate is 5e-4. The label smoothing rate is set to 0.1, the warm-up steps to 4000, and the dropout to 0.3. Adam optimizer (Kingma and Ba, 2015) with $\beta1$=0.9 and $\beta2$=0.98 is also used. In the evaluation phase, we use Marian[4] (Junczys-Dowmunt et al., 2018) for decoding and then calculate the sacreBLEU scores on the WMT22 chat translation task dev sets to measure the performance of each model.

System	22 en→de sent	22 en→de doc	22 de→en sent	22 de→en doc
baseline	50.3	-	58.7	-
+ Data Selection	62.5	-	65.5	-
+ Bilingual Doc	36.7	37.1	45.2	44.5
+ Bilingual & Doc bt	54.6	55.6	56.8	56.3

Table 4: The results of different strategies in the document-level model. Bilingual Doc means using WMT news bilingual doc data to finetune previous model. Bilingual & Doc bt means using WMT news bilingual doc data and pseudo corpus generated from WMT news monolingual doc data to finetune previous model.

System	sacrebleu↑	total↑	er↑	es↑	sie↑
Baseline	25.8	0.37	0.12	0.82	0.16
+ Data Selection	**26.3**	0.36	0.13	0.81	0.14
+Bilingual Doc	23.0	**0.41**	0.19	0.69	0.34
+Bilingual Doc & DOC BT	24.7	0.36	0.11	0.82	0.15

Table 5: The higher the accuracy of pronoun translation, the better the model combines contextual information.

5 Result and Analysis

Table 3 shows the main results on the development sets. Bilingual data selection gains significant improvement on dev sets. Although bilingual data is selected based on dev sets, the selected data consists of 13M sentences. Therefore, there is no overfitting risk. This strategy also improves model performance on chat20 test sets.

Since the dev set is already used to select data, we no longer use the dev set to fine-tune model. We then continue to train our model using noisy self-training strategy on monolingual in-domain data. The result shows that there is an increase in BLEU on the en→de track. After model averaging, performance on de→en track improves, but performance on en→de deteriorates.

Finally, we select the result of data selection after model averaging as the primary submission, noisy self-training after model averaging as the contrastive2. Note that, the two submissions before are few-shot. The result of the baseline model as the submission of the zero-shot track, is contrastive1.

5.1 Document-level NMT

According to the test results shown in Table 4, using document-level data to optimize models does not work well, mostly because this data is from the news domain. From our subsequent experiments, we also find that chat tasks have high requirements on data domain.

From rows 4 and 5 in Table 4, the model using bilingual document data has worse results than the model using DocBT data. We assume that there are two reasons for this phenomenon. One is that the size of bilingual documents is limited, and the other is that the DocBT data generated using the data selection model is closer to the chat domain than the original bilinguals.

To verify the effectiveness of our document-level translation model, we evaluate our model on (Müller et al., 2018) test set, which is a pronoun translation accuracy task.

As shown in Table 5, the pronoun translation accuracy of bilingual document-level model was significantly better than that of other models. But the BLEU is the lowest due to the minimum amount of data. From subsequent domian transfer experiments, we can also find, chat tasks are extremely sensitive to the domain of the data, but we cannot find enough chat data to train the document-level translation model. Therefore, we cannot draw a conclusion that document-level translation is useless for chat translation tasks. Further researches can be carried out when sufficient chat data is available.

5.2 Domain Transfer

Since no chat training data is provided except for the development set, we continue to train the baseline model using development set and monolingual data from previous chat tasks. As shown in rows 3 and 4 in the Table 6, models training with chat20 development set performs well on the chat20 test set. However, little improvement is observed on chat22 dev set. As mentioned above, the data dis-

[3] https://github.com/facebookresearch/fairseq
[4] https://github.com/marian-nmt/marian

System	20 en→de test	20 de→en test	22 en→de dev	22 de→en dev
baseline	45.1	46.7	50.3	58.7
+ 20 dev fine-tune	60.5	63.5	52.4(+1.9)	53.5(-5.1)
+ 20 mono en FT/BT	59.6	64.1	45.7(-4.6)	31.6(-27.1)
+ Subtitle en FT/BT	57.1	53.5	43.7(-6.6)	28.5(-30.2)
+ Conversation en FT/BT	56.7	51.4	49.5(-0.5)	43.8(-14.9)
+ Shopping en FT/BT	56.2	55.9	50.3(-)	43.3(-15.4)
+ Data Selection	49.2	48.1	62.5(+12.2)	65.5(+6.8)

Table 6: The results of different strategies for the sentence-level model. FT/BT means that forward translation in en→de direction and back translation in de→en direction.

tribution for these two tasks is not consistent.

Monolingual data of similar domains, such as subtitle, conversation, and shopping, is then used for FT or BT enhancement. From rows 5, 6 and 7 in the Table 6, the results are worse than using chat20 data. Although the monolingual data is of higher quality, its domain and style are far away from chat data. So it brings no improvement.

5.3 Bilingual Data Selection

Through the above experiments, we find that this year's chat task has unique features and is very sensitive to domain differences. Using the idea proposed by Wang et al. (2019, 2018), we select 13M data from 40M general bilingual data to optimize our baseline model.

As can be seen from row 8 in the Table 6, this strategy is effective and improves the translation quality in both directions. Besides, we find that the data selected using the chat22 development set also improves model performance on the chat20 task, indicating that this strategy is a general method. We will test its applicability in the future.

6 Conclusion

This paper presents the submissions of HW-TSC to the WMT 2022 Chat Translation Shared Task. For both direction in customer-agent translation task, we perform experiments with a series of preprocessing and training strategies. The results show that bilingual data selection achieves the best results. In the future, we will continue to explore the applicability of bilingual data selection mentioned in this paper.

Besides, the performance of document-level translation model is limited given the amount of data. It has not achieved the expected results on this task, and we will continue to explore the impact of context for the chat task.

References

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Guangsheng Bao, Yue Zhang, Zhiyang Teng, Boxing Chen, and Weihua Luo. 2021. G-transformer for document-level machine translation. In *ACL*.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Daniel Duckworth, Semih Yavuz, Ben Goodrich, Amit Dubey, Andy Cedilnik, and Kyu-Young Kim. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. In *EMNLP*.

Isaac Caswell, Ciprian Chelba, and David Grangier. Tagged back-translation. *WMT 2019*, page 53.

P Dakwale and C Monz. 2017. Fine-tuning for neural machine translation with limited degradation across in-and out-of-domain data.

Carsten F. Dormann, Justin M. Calabrese, Gurutzeta Guillera-Arroita, Eleni Matechou, Volker Bahn, Kamil Bartoń, Colin M. Beale, Simone Ciuti, Jane Elith, Katharina Gerstner, Jérôme Guelat, Petr Keil, José J. Lahoz-Monfort, Laura J. Pollock, Björn Reineking, David R. Roberts, Boris Schröder, Wilfried Thuiller, David I. Warton, Brendan A. Wintle, Simon N. Wood, Rafael O. Wüest, and Florian Hartig. 2018. Model averaging in ecology: a review of bayesian, information-theoretic, and tactical approaches for predictive inference. *Ecological Monographs*, 88(4):485–504.

Zi-Yi Dou, Zhaopeng Tu, Xing Wang, Shuming Shi, and Tong Zhang. 2018. Exploiting deep representations for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4253–4262.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

M. Amin Farajian, António V. Lopes, André F. T. Martins, Sameen Maruf, and Gholamreza Haffari. 2020. Findings of the WMT 2020 shared task on chat translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 65–75, Online. Association for Computational Linguistics.

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 56–60.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252.

Miguel Graça, Yunsu Kim, Julian Schamper, Shahram Khadivi, and Hermann Ney. 2019. Generalizing back-translation in neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 45–52.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2020. Revisiting self-training for neural sequence generation. In *Proceedings of ICLR*.

Kenji Imamura and Eiichiro Sumita. 2018. Nict self-training approach to neural machine translation at nmt-2018. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 110–115.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Herve Jegou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models.

Armand Joulin, Édouard Grave, Piotr Bojanowski, and Tomáš Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at wmt 2019: Towards large-scale document-level neural machine translation. In *WMT*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, et al. 2018. Marian: Fast neural machine translation in c++. In *ACL (4)*.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83.

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *ACL (1)*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP (Demonstration)*.

Zhiyi Ma, Sergey Edunov, and Michael Auli. 2021. A comparison of approaches to document-level machine translation. *ArXiv*, abs/2101.11040.

Mathias Müller, Annette Rios, Elena Voita, and Rico Sennrich. 2018. A Large-Scale Test Set for the Evaluation of Context-Aware Pronoun Translation in Neural Machine Translation. In *WMT 2018*, Brussels, Belgium. Association for Computational Linguistics.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *ICML*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*, pages 3104–3112.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. Context-aware monolingual repair for neural machine translation. In *EMNLP*.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting cross-sentence context for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2826–2831, Copenhagen, Denmark. Association for Computational Linguistics.

Wei Wang, Isaac Caswell, and Ciprian Chelba. 2019. Dynamically composing domain-data selection with clean-data selection by “co-curricular learning” for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1282–1292, Florence, Italy. Association for Computational Linguistics.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018. Denoising neural machine translation training with trusted data and online data selection. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 133–143, Brussels, Belgium. Association for Computational Linguistics.

Daimeng Wei, Zongyao Li, Zhanglin Wu, Zhengzhe Yu, Xiaoyu Chen, Hengchao Shang, Jiaxin Guo, Minghan Wang, Lizhi Lei, Min Zhang, Hao Yang, and Ying Qin. 2021. HW-TSC’s participation in the WMT 2021 news translation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 225–231, Online. Association for Computational Linguistics.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4207–4216.

Lei Yu, Laurent Sartran, Wojciech Stokowiec, Wang Ling, Lingpeng Kong, Phil Blunsom, and Chris Dyer. 2019. Putting machine translation in context with the noisy channel model. *ArXiv*, abs/1910.00553.

Clean Text and Full-Body Transformer: Microsoft's Submission to the WMT22 Shared Task on Sign Language Translation

***Subhadeep Dey, Abhilash Pal, Cyrine Chaabani, *Oscar Koller**
Microsoft - Munich, Germany
{subde,t-apal,t-cchaabani,oskoller}@microsoft.com

Abstract

This paper describes Microsoft's submission to the first shared task on sign language translation at WMT 2022, a public competition tackling sign language to spoken language translation for Swiss German sign language. The task is very challenging due to data scarcity and an unprecedented vocabulary size of more than 20k words on the target side. Moreover, the data is taken from real broadcast news, includes native signing and covers scenarios of long videos. Motivated by recent advances in action recognition, we incorporate full body information by extracting features from a pre-trained I3D model and applying a standard transformer network. The accuracy of the system is further improved by applying careful data cleaning on the target text. We obtain BLEU scores of **0.6** and 0.78 on the test and dev set respectively, which is the best score among the participants of the shared task. Also in the human evaluation the submission reaches the first place. The BLEU score is further improved to **1.08** on the dev set by applying features extracted from a lip reading model.

1 Introduction

Sign languages are natural visual languages that are used by deaf and hard of hearing individuals to communicate in everyday life. Sign languages are actively being researched. However, there is a huge imbalance in the field of natural language and speech processing between oral and signed languages. Since recently, one observes the emergence of a transition shifting sign language processing to be part of the NLP mainstream (Yin et al., 2021). We embrace this development which manifests (among others) in the creation of the first shared task on sign language translation as part of WMT 2022 (Mathias et al., 2022). It is great to have real-world sign language data (Bragg et al., 2019; Yin et al., 2021) as the basis of this shared task, manifested in native signers content and an unprecedentedly large vocabulary. Nevertheless, this leads to a very challenging task with low performance numbers. When participating in the advances of sign language technologies it is worth recapping that deaf people have much at stake, both to gain and lose from applications that will be enabled here (Bragg et al., 2021). We aim to advance the field and the use-cases in a positive way and present our findings in this system paper.

In the remainder of this work we first present a brief view on the relevant literature in Section 2, then we present the employed data in Section 3. Subsequently, we describe our submission in Section 4, additional experiments in Section 5 and we end with a summary in Section 6.

2 Related Work

In this section, we present a limited overview of related work in sign language translation. We focus this review on the translation direction from sign language to spoken language and dismiss approaches that target the opposite direction, i.e. sign language production.

Sign language translation started targeting written sign language gloss to spoken language text translations, hence no videos were involved. Related works were mainly based on phrase-based systems employing different sets of features (Stein et al., 2007, 2010; Schmidt et al., 2013). Then, neural machine translation revolutionized the field. The first research publications on neural sign language translation were based on LSTMs either with full image input (Camgoz et al., 2018) or utilized human keypoint estimation (Ko et al., 2019). Transformer models then replaced the recurrent architectures (Camgoz et al., 2020; Yin and Read, 2020; Yin, 2020). These models perform a lot better, but suffer from a basic drawback that the input sequences must be limited to a maximum length. Previous work (Camgoz et al., 2018; Or-

*Equal Contribution

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 969 - 976
December 7-8, 2022 ©2022 Association for Computational Linguistics

bay and Akarun, 2020) has identified the need for strong tokenizers to produce compact representations of the incoming sign language video footage. Hence, a considerable body of publications target creating tokenizer models that are often trained on sign language recognition data sets (Koller et al., 2020, 2016; Zhou et al., 2022) or sign spotting data sets (Albanie et al., 2020; Varol et al., 2021; Belissen et al., 2019; Pfister et al., 2013).

There are several data sets relevant for sign language translation. Some of the most frequently encountered are RWTH-PHOENIX-Weather 2014T (Koller et al., 2015a; Camgoz et al., 2018) and the CSL (Huang et al., 2018) (which could be also considered a recognition data set). However, there are promising new data sets appearing: OpenASL (Shi et al., 2022a), SP-10 dataset (Yin et al., 2022) (covers mainly isolated translations) and How2Sign (Duarte et al., 2021).

3 Data

To train our system, we used the training data provided by the shared task organizers. The data can be considered real-life-authentic as it stems from broadcast news using two different sources: FocusNews and SRF. FocusNews, henceforth FN, is an online TV channel covering deaf signers with videos of 5 minutes having variable sampling rates of either 25, 30 or 50 fps. SRF represents public Swiss TV with contents from daily news and weather forecast which are being interpreted by hearing interpreters. The videos are recorded with a sampling rate of 25 fps. All data, therefore, covers Swiss German sign language (DSGS). Our feature extractors are pretrained on BSL-1k (Albanie et al., 2020) and AV-HuBERT (Shi et al., 2022b). Additionally, we evaluate the effect of introducing a public sign language lexicon that covers isolated signs [1], which we refer to as Lex. It provides main hand shape annotations, one or multiple (mostly one) examples of the sign and an example of how this sign is used in a continuous sentence. We choose a subset that overlaps in vocabulary with either FocusNews or SRF. As part of the competition, independent dev and test sets are provided, which consist of 420 and 488 utterances respectively.

Table 1 shows the statistics of the training data. We see, that there is about 35 hours of training data in total. In raw form without any preprocessing the data is case sensitive, contains punctuation and digits. In this raw form the vocabulary amounts to close to 35k different words on the target side (which is written German). 22k words of these just occur a single time in the training data (singletons). Through careful preprocessing as described in Section 4.3 we can shrink the vocabulary to about 22k words and the singletons to about 12k.

	SRF	**FN**	**Lex**	**Total**
Videos	29	197	1201	1427
Hours	15.6	19.1	0.9	35.6
Raw: no preprocessing				
Vocabulary	18942	21490	–	34783
Singletons	12433	13624	–	22083
Clean: careful preprocessing				
Vocabulary	13029	14555	821	22840
Singletons	7483	7923	591	12290

Table 1: Data statistics on data used for training. SRF and FN refer to SRF broadcast and FocusNews data, while Lex stands for a public sign language lexicon. Singletons are words that only occur a single time during training.

4 Submitted System

Sign languages convey information through the use of manual parameters (hand shape, orientation, location and movement) and non-manual parameters (lips, eyes, head, upper body). To capture most information from the signs, we opt for an RGB-based approach, neglecting the tracked skeleton features by the shared task organizers. For the submitted system we rely on a pre-trained tokenizer for feature extraction and train a sequence-to-sequence model to produce sequences of whole words (no byte pair encoding). We further pre-process the sentences (ground truths of the videos) to clean it. This step is crucial to push the model to focus more on semantics of the data. Finally, in order to adhere to the expected output format for the submission, we convert the text back to display format using Microsoft's speech service. This applies inverse text normalization, capitalization and punctuation to the output text to make it more readable. The details of various components of the system are described in the next subsections.

4.1 Features

We use a pre-trained I3D (Carreira and Zisserman, 2017) model, based on inflated inceptions with 3D convolutional neural networks, to extract features

[1] https://signsuisse.sgb-fss.ch/

for our task. The model (Varol et al., 2021) was pre-trained to take consecutive video frames as input and predict over 1k sign classes. It was trained on BSL-1K (Albanie et al., 2020) consisting of about 700k spotted sign instances from the British broadcast news. The features are extracted with a context window of 64 frames and a temporal stride of 8. We use the model as a feature extractor, recovering embeddings before the final classification layer (layer: mixed_5c), yielding a sequence of 1024 dimensional vector, extracted for each video. Our input data is required to match the training conditions of the network, hence we apply gray background padding (adding 20% padding left and right, 7.5% up and down) and rescale the videos to 224 × 224 resolution. The front end features are subsequently fed to a sequence model.

4.2 Sequence model

A standard transformer network is trained to predict text sequences. We apply word-based units as the output instead of byte pair encoding. It seems that full words help reduce ambiguity in a data constrained scenario. The model is trained with the fairseq (Ott et al., 2019) toolkit. We apply 3 transformer layers at the encoder side and 2 layers at the decoder with 1024 hidden feed-forward dimension and 22k output units. The model is trained with Adam optimizer for 2k epochs with a learning rate of 1e-3. We found that a beam size of 1 works well on the dev set during decoding.

4.3 Data Cleaning

To reduce ambiguity and noise in the target text, we first applied manual cleaning by removing foreign (French and English) sentences. We then proceeded to removing sentences that start with a hashtag sign, as this seems to indicate inaccurate annotation. Further, we removed status messages that were added by the subtitling agency (such as "1:1-Untertitelung.", "Livepassagen können Fehler enthalten.", "Mit Live-Untertiteln von SWISS TXT") and patterns enclosed by an asterisk which indicate sounds occuring in the show (e.g. "* Beschwingte Blasmusik *"). As a next step, we expanded abbreviations like "Mrd." to "Milliarden" and applied text normalization to remove punctuation and special characters, lower case of the text and expand numbers and dates. As can be shown in Table 1, this plays a major role in reducing the total vocabulary.

Data cleaning	Dev (RedB)	Dev	Test
no	0.49	0.70	0.4
yes	**0.78**	**0.77**	**0.6**

Table 2: The Table shows the effect of data cleaning. Performance of translation systems trained on SRF and FocusNews evaluated on the WMT 2022 dev and test data is provided. We report reduced and standard BLEU score on the Dev set and only standard BLEU on the Test set. RedB stands for the reduced BLEU measure.

4.4 Evaluation metrics

A common evaluation metric for machine translation is BLEU (Papineni et al., 2002). However, the difficulty of the given task and the inherently low performance of the submitted systems cause a bias in the automated evaluation. Spoken languages like Swiss German, which is the target output space for the translation in this challenge, follow a statistical pattern where stop words or function words constitute the classes of words that occur most frequently. This explains one of the observations that we made in regard to the generated system output. In fact, the models which were producing more stop words achieved the highest BLEU scores. For example, the model we submitted achieved a BLEU score of 0.77 on the dev set, while an earlier, clearly worse, checkpoint achieved 0.91. Looking at the stop words, the submitted model output counts 2125 stop words, while the earlier checkpoint counts 2237 of such words. After our stop word removal the submitted reduced BLEU score is 0.78, while the earlier model achieves only 0.66. Hence, for model selection, we propose to use 'reduced BLEU' with an additional step: we first apply a blacklist to remove stop and function words. Using this approach, the model selection process based on the reduced BLEU metric turned to be much more reliable and more reflective of actual performance. In this work, we report both reduced BLEU and standard BLEU for all results on the dev set. Only standard BLEU is reported by the automatic evaluation through the shared task on the test set. The list of employed stop words can be found in the appendix.

4.5 Results

The results of the submitted systems are presented in Table 2, which allows to compare the effect of applying data cleaning. In terms of reduced BLEU, data cleaning improves the performance from 0.49

to 0.78 on the dev data. The test data shows similarly an improvement from 0.4 BLEU to 0.6. Based on the preliminary automatic BLEU scoring this result was the best among the participants of the shared task. However, it has to be noted that the final evaluation will be based on a human eval that has not yet been completed at the time of paper submission. It can be concluded that careful data preparation is fundamental for this data. Nevertheless, the shared task proves to be very challenging and overall, we observe rather low performance compared to published results on benchmark data sets like RWTH-PHOENIX 2014T (Camgoz et al., 2018). This further proves the large amount of variability in the task, which is amplified due to the presence of high numbers of singleton words.

5 Additional Experiments

We perform additional experiments to assess the impact of having dedicated mouth features as well as a lexicon data set on the model performance. The mouth carries important semantic information in sign languages. In the literature, exploiting lip information has shown to be fundamental for increased performance in sign language recognition and translation (Koller et al., 2015b; Shi et al., 2022a).

5.1 Mouth features

To extract features from the lip area, we employ a pre-trained AV-HuBERT model (Shi et al., 2022b) that has been trained on English data. AV-HuBERT is trained to learn a robust audio-visual representation in a self-supervised fashion. The success of the model is evident by the performance on an audio-visual speech recognition task. It has proven to be useful for sign-language task as well (Shi et al., 2022a). For us, the first step is to obtain mouth patches from the video frames. Hence, we rely on the dlib utility provided by the AV-HuBert authors for obtaining facial key point extraction. The face patch is cropped and re-scaled to match the input size (96 x 96) of the model. We use the AV-HuBERT model to extract 768 dimensional embedding (output of the ResNet layer) to obtain a feature vector per frame.

5.2 Comparison to RWTH-PHOENIX 2014 T

To underline the difficulty of the given shared task, we compare our employed pipeline on a standard benchmark data set for sign language translation, namely Phoenix 2014T (Camgoz et al., 2018). We noticed a small difference between the original PHOENIX 2014T corpus as referenced and shared in of (Camgoz et al., 2018) and the publicly available embeddings and experiments of (Camgoz et al., 2020). In the latter full stops mark the end of each utterance, while in the original version this is not the case. The effect is small, but for the sake of completeness, we show it here. Furthermore, considering the statistics of PHOENIX 2014T and this WMT 2022 shared task, the difference becomes apparent. Table 3 shows the key statistics for the two tasks side by side. We can see that the WMT 2022 task has a nearly 8 times larger vocabulary, with 11 times more singletons that occur only once in training. However, it has not even 4 times more video material.

	WMT 2022	PHOENIX 2014T	Factor
Hours	35.6	9.2	3.9
Vocabulary	22840	2887	7.9
Singletons	12290	1077	11.4

Table 3: Comparison between the training data for the WMT 2022 shared task and PHOENIX 2014T.

	Training data	Features	
		Full body	Mouth
RedB	SRF + FN	0.78	0.95
	SRF + FN + Lex	0.54	1.08
Stand.	SRF + FN	0.77	1.15
	SRF + FN + Lex	0.68	1.27

Table 4: The effect of adding lexicon data to systems trained with full body features (I3D) and mouth features (AV-HuBERT). Configurations are evaluated on the WMT 2022 Dev dataset using the reduced BLEU (RedB) and standard BLEU (Stand.) score as metric.

5.3 Results

Table 4 shows the results of applying AV-HuBERT features as input to the sequence model. It can be observed that the model trained with AV-HuBERT features performs better than the I3D model. In fact, it achieves a 0.95 reduced BLEU score, while the full body I3D features reach only 0.78. On visual inspection, we found that the model trained with AV-HuBERT is able to predict infrequent words but fails on simple words such as "auf wiedersehen". Therefore, we assess the effect of adding a lexical data set (‘Lex’) to boost representations of those simple words. Table 4 shows that the addition of lexical data further improves the model performance which reaches 1.08 reduced BLEU. We believe that this is likely due to the matching lip movement patterns in the lexical training dataset and dev dataset. Unfortunately, due to time constraints, we were not able to submit this model.

Table 5 shows results on PHOENIX 2014T. We can see that our submitted pipeline matches the performance of (Camgoz et al., 2020) (19.80/20.24 BLEU on the dev/test sets compare to 20.69/20.17). However our employed embeddings, which were not trained on PHOENIX 2014T, do not generalize well to PHOENIX 2014T and are hence significantly outperformed by the ones employed in (Camgoz et al., 2020). The experiments show that the WMT 2022 shared task is significantly more challenging than PHOENIX 2014T. We also see that the addition of full stops at the end of each utterances in PHOENIX 2014T amounts to a difference in BLEU of about 1% relative (14.22/13.22 BLEU with full stops opposed to 14.06/13.13 without full stops).

6 Summary

In this paper, full body information is applied successfully for a challenging sign language task as part of the WMT 2022 competition. As such, we employed a pre-trained I3D model to extract an embedding for a sequence of frames of the video. The features are further fed as input to a standard transformer network. We obtain reasonable performance of 0.4 in terms of BLEU score on the test set. The model is further enhanced by applying careful cleaning to the text output. We obtain the result of 0.6 BLEU score on the official test data. Based on the automatic BLEU scoring this result was the best among the 7 participants of the shared task, but also in the human evaluation our submission reaches the first place. With additional experiments, we validate the usefulness of a pre-trained lip reading model for this task and the addition of a lexical data set. This improves the results to 1.08 reduced BLEU on the dev set.

Limitations

One major limitation to our work resides within the data set used for training our model. In fact, the signing interpreter is usually not a native signer and often seems to be heavily influenced by the source language, a.k.a the spoken language. As stated previously, we used a signing interpreter for SRF data. Another issue we have identified lies in the limited domain of the data, as it is constrained to Broadcast news. The trained model may therefore be too specialized to generalize well beyond this area. Furthermore, due to the small number of individuals present in the data set, it remains unclear if and how much ethnicity bias is introduced to the model. Our team did not proceed with any experiments to identify and measure this. However, we do believe that it is crucial to further analyze possible biases in the future. One evaluation metric that we did take into consideration for the model performance is the BLEU score. The experiments consistently returned extremely low values which reflects a poor accuracy. One thing that remains unclear to us is how significant small BLEU differences are for human perception and subjective evaluation.

References

Samuel Albanie, Gül Varol, Liliane Momeni, Triantafyllos Afouras, Joon Son Chung, Neil Fox, and Andrew Zisserman. 2020. BSL-1K: Scaling up co-articulated sign language recognition using mouthing cues. In *European Conference on Computer Vision*.

Valentin Belissen, Michèle Gouiffès, and Annelies Braffort. 2019. Automatic recognition of Sign Language structures in RGB videos: The detection of pointing and lexical signs.

Danielle Bragg, Naomi Caselli, Julie A. Hochgesang, Matt Huenerfauth, Leah Katz-Hernandez, Oscar Koller, Raja Kushalnagar, Christian Vogler, and Richard E. Ladner. 2021. The FATE Landscape of Sign Language AI Datasets: An Interdisciplinary Perspective. *ACM Transactions on Accessible Computing*, 14(2):7:1–7:45.

Danielle Bragg, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudreault, Annelies Braffort, Naomi Caselli, Matt Huenerfauth, Hernisa Kacorri, Tessa Verhoef, Christian Vogler, and Meredith Ringel Morris. 2019. Sign Language Recognition, Generation, and Translation: An Interdisciplinary Perspective. In *Proc. Int. ACM SIGACCESS Conf. on Computers and Accessibility (ASSETS)*, ASSETS '19, pages 16–31, New York, NY, USA.

Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural Sign Language Translation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 7784–7793, Salt Lake City, UT.

Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020. Sign Language Trans-

Approach	Decoding	Fullstops		PHOENIX 2014T	
	Beam	No	Yes	Dev	Test
(Camgoz et al., 2020)	10		x	20.69	20.17
Our pipeline with embeddings from (Camgoz et al., 2020)	5		x	19.80	20.24
Our pipeline with WMT full body embeddings	5	x		14.06	13.13
Our pipeline with WMT full body embeddings	5		x	14.22	13.22

Table 5: Showing results on PHOENIX 2014T, a benchmark data set. The system which matches our submission to the shared task can be found on the last line.

formers: Joint End-to-End Sign Language Recognition and Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10023–10033.

Joao Carreira and Andrew Zisserman. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308.

Amanda Duarte, Shruti Palaskar, Lucas Ventura, Deepti Ghadiyaram, Kenneth DeHaan, Florian Metze, Jordi Torres, and Xavier Giro-i-Nieto. 2021. How2Sign: A Large-Scale Multimodal Dataset for Continuous American Sign Language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2735–2744.

Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li. 2018. Video-based sign language recognition without temporal segmentation. In *Proc. of the AAAI Conf. on Artificial Intelligence*, pages 2257–2264, New Orleans, Louisiana, USA.

Sang-Ki Ko, Chang Jo Kim, Hyedong Jung, and Choongsang Cho. 2019. Neural Sign Language Translation Based on Human Keypoint Estimation. *Applied Sciences*, 9(13):2683.

Oscar Koller, Necati Cihan Camgoz, Hermann Ney, and Richard Bowden. 2020. Weakly Supervised Learning with Multi-Stream CNN-LSTM-HMMs to Discover Sequential Parallelism in Sign Language Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(09):2306–2320.

Oscar Koller, Jens Forster, and Hermann Ney. 2015a. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding (CVIU)*, 141:108–125.

Oscar Koller, Hermann Ney, and Richard Bowden. 2015b. Deep Learning of Mouth Shapes for Sign Language. In *Proc. IEEE Int. Conf. on Computer Vision Workshops (ICCVW)*, pages 477–483, Santiago, Chile.

Oscar Koller, Hermann Ney, and Richard Bowden. 2016. Deep Hand: How to Train a CNN on 1 Million Hand Images When Your Data Is Continuous and Weakly Labelled. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3793–3802, Las Vegas, NV, USA.

Müller Mathias, Sarah Ebling, Avramidis Eleftherios, Alessia Battisti, Michèle Berger, Richard Bowden, Annelies Braffort, Necati Cihan Camgöz, Cristina España-Bonet, Roman Grundkiewicz, Zifan Jiang, Oscar Koller, Amit Moryossef, Regula Perrollaz, Sabine Reinhard, Annette Rios, Dimitar Shterionov, Sandra Sidler-Miserez, Katja Tissi, and Davy Van Landuyt. 2022. Findings of the WMT 2022 shared task on sign language translation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alptekin Orbay and Lale Akarun. 2020. Neural Sign Language Translation by Learning Tokenization. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 222–228.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. Fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Tomas Pfister, James Charles, and Andrew Zisserman. 2013. Large-scale learning of sign language by watching TV (using co-occurrences). In *Proc. British Machine Vision Conference (BMVC)*, pages 1–11, Bristol, UK.

Christoph Schmidt, Oscar Koller, Hermann Ney, Thomas Hoyoux, and Justus Piater. 2013. Using Viseme Recognition to Improve a Sign Language Translation System. In *International Workshop on Spoken Language Translation*, pages 197–203, Heidelberg, Germany.

Bowen Shi, Diane Brentari, Greg Shakhnarovich, and Karen Livescu. 2022a. Open-Domain Sign Language Translation Learned from Online Video.

Bowen Shi, Wei-Ning Hsu, Kushal Lakhotia, and Abdelrahman Mohamed. 2022b. Learning Audio-Visual Speech Representation by Masked Multimodal Cluster Prediction. In *International Conference on Learning Representations*.

Daniel Stein, Philippe Dreuw, Hermann Ney, Sara Morrissey, and Andy Way. 2007. Hand in Hand: Automatic Sign Language to Speech Translation. In *Proc. Conf. on Theoretical and Methodological Issues in Machine Translation*, pages 214–220, Sk"ovde, Sweden.

Daniel Stein, Christoph Schmidt, and Hermann Ney. 2010. Sign Language Machine Translation Overkill. In *International Workshop on Spoken Language Translation*, pages 337–344, Paris, France.

Gul Varol, Liliane Momeni, Samuel Albanie, Triantafyllos Afouras, and Andrew Zisserman. 2021. Read and Attend: Temporal Localisation in Sign Language Videos. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16852–16861, Nashville, TN, USA. IEEE.

Aoxiong Yin, Zhou Zhao, Weike Jin, Meng Zhang, Xingshan Zeng, and Xiaofei He. 2022. MLSLT: Towards Multilingual Sign Language Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5109–5119.

Kayo Yin. 2020. Sign Language Translation with Transformers. *arXiv:2004.00588 [cs]*.

Kayo Yin, Amit Moryossef, Julie Hochgesang, Yoav Goldberg, and Malihe Alikhani. 2021. Including Signed Languages in Natural Language Processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7347–7360, Online. Association for Computational Linguistics.

Kayo Yin and Jesse Read. 2020. Better Sign Language Translation with STMC-Transformer. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5975–5989, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Hao Zhou, Wengang Zhou, Yun Zhou, and Houqiang Li. 2022. Spatial-Temporal Multi-Cue Network for Sign Language Recognition and Translation. *IEEE Transactions on Multimedia*, 24:768–779.

A List of Stop Words for Reduced BLEU Estimation

- ab
- als
- als
- also
- am
- am
- an
- an
- andere
- auf
- aus
- beim
- bin
- bist
- da
- darauf
- das
- dass
- davon
- dazu
- dem
- den
- denen
- der
- des
- des
- deshalb
- dessen
- die
- dies
- diese
- diesen
- dieser
- dieses
- doch
- dort
- ein
- eine
- einem
- einen
- einen
- einer
- eines
- eines
- er
- es
- es
- für
- gar
- gegen
- geht’s
- genau
- gibt
- habe
- haben
- habt
- hast
- hast
- hat
- hat
- hatte
- hätte
- hatten
- hätten
- her
- hin
- ihm
- ihre
- ihre
- im
- in
- ins
- ist
- könne
- könnte
- könnten
- man
- mehr
- mit
- noch
- nun
- ob
- oder
- quasi
- schon
- sehr
- sei
- seid
- seien
- sein
- seit
- sich
- sie
- sie
- sind
- so
- solchen
- soll
- somit
- sowie
- sowohl
- statt
- über
- um
- und
- vom
- von
- vor
- war
- war
- wäre
- war’s
- wars
- warst
- wart
- wegen
- weiteren
- weiterhin
- wem
- wen
- wenn
- werde
- werden
- werdet
- weshalb
- wie
- will
- wir
- wird
- wirst
- wo
- wohl
- wolle
- wollte
- wollten
- worauf
- wurde
- würde
- würden
- zu
- zudem
- zum
- zur
- zur
- zur
- zwar

DFKI-MLT at WMT-SLT22
Spatio-temporal Sign Language Representation and Translation

Yasser Hamidullah and **Josef van Genabith** and **Cristina España-Bonet**
{yasser.hamidullah,Josef.van_Genabith,cristinae}@dfki.de
German Research Center for Artificial Intelligence (DFKI)
Saarland Informatics Campus, Saarbrücken, Germany

Abstract

This paper describes the DFKI-MLT submission to the WMT-SLT 2022 sign language translation (SLT) task from Swiss German Sign Language (video) into German (text). State-of-the-art techniques for SLT use a generic seq2seq architecture with customized input embeddings. Instead of word embeddings as used in textual machine translation, SLT systems use features extracted from video frames. Standard approaches often do not benefit from temporal features. In our participation, we present a system that learns spatio-temporal feature representations and translation in a single model, resulting in a real end-to-end architecture expected to better generalize to new data sets. Our best system achieved 5 ± 1 BLEU points on the development set, but the performance on the test dropped to 0.11 ± 0.06 BLEU points.

1 Introduction

Text-to-text machine translation (MT) is achieving a great success with even (close to) human performance for some language pairs and domains (Akhbardeh et al., 2021). However, the situation in sign language translation (SLT) is much different. One important reason is that the SLT is a low-resource scenario where one does not have the same amount of data as in high-resourced text-to-text to achieve a similar level of performance. A more specific reason is that SLT involves two modalities, text and video. Various problems arise when dealing with these modalities. Besides data scarcity, the lack of temporal boundaries in the input videos is a challenge. To overcome the lack of temporal boundaries, the most common solution tends to ignore or not benefit from temporal features. This approach relies on the Transformer (Vaswani et al., 2017) capabilities to learn sequence-to-sequence tasks. The state-of-the-art SLT technique (Camgöz et al., 2020) is practically a normal Transformer but uses a custom embedding layer for 2D features extracted from video frames. In this approach, training a SLT system requires a pre-extraction step to convert frame features to vectors and train a Transformer separately to translate the vectors into spoken language. This type of approach has been widely used on a very specific dataset, the weather forecast corpus PHOENIX14T (Camgoz et al., 2018), where researchers reported a relatively good performance in terms of BLEU (∼20) (Camgöz et al., 2020; Min et al., 2021).

Despite its good performance on a specific dataset, there is the doubt whether such type of architecture generalizes to new data sets. In order to build a more general technique, we focus on fundamental SLT problems such as the design, implementation and evaluation of a fully end-to-end model and representation learning for sign language videos. Having a fully end-to-end model facilitates the task of data collection and diminishes the need for annotation (e.g. in terms of sign language glosses), which is necessary to build larger and richer datasets. It also allows training video embeddings fully optimized for the translation task. Text translation is one of the most mature areas in natural language processing, and therefore we focus here on the sign language representation part of the architecture and use an in-house state-of-the-art Transformer for text generation.

This paper reports our approach for end-to-end SLT used for the WMT-SLT translation shared task from Swiss German Sign Language into German. In the next sections we introduce our approach (Section 2), experimental setup (Section 3), results (Section 4) and conclusions & perspectives (Section 5).

2 Our Approach

The main idea of our approach is to learn feature representation and translation in a single model, and be able to train them together. Figure 1(a) sketches our general pre-processing pipeline and

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 977 - 982
December 7-8, 2022 ©2022 Association for Computational Linguistics

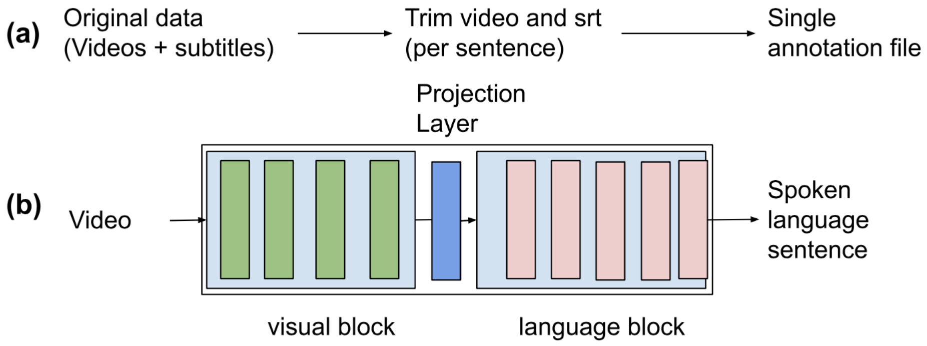


Figure 1: End-to-end architecture for sign language translation.

Figure 1(b) the architecture. The system architecture consists of two connected blocks: the first block made of CNNs is intended for vision and the second one is for language which is Transformer architecture.

Both the CNNs used for the video representations and the Transformers used for the text representations come with large numbers of parameters. As we are operating in a low resource scenario, besides the combination of the two networks, we experiment to find the best trade-off between data size and number of parameters.

2.1 Visual feature representation

Our goal is to build a sentence embedding-like model for the visual sign language encoder, as a word/sign level-like representation is limited by the lack of temporal boundaries in videos. We hypothesize that a sentence embedding will still contain and distinguish all the information given by individual signs.

In the shared task, we use ResNet3D (Hara et al., 2017) as our spatio-temporal visual feature representation block. We prefer it instead of a normal 2D with temporal convolutions (Wang et al., 2019) to develop a fully end-to-end trainable model. There are many available architectures in the literature, but ResNet is unique in providing different models with different scales. This gives us the possibility to experiment with various sizes which help us to weight the importance of each of the vision and text blocks in our trade-off experiments.

Our visual encoding in the submitted system is composed by the original 3D ResNet10 with output conversion. The conversion creates a sequence of vectors from the single output vector to adapt to the transformer encoder input. We define the **SWM** parameter (Sentence to Words Mapping), which is the number of splits from the output vector. This output is projected through a linear layer which is connected directly to the language block. We experiment with 3D ResNet10, 3D ResNet34 and 3D ResNet50 and show the comparative results in Section 3.

2.2 Language representation

The language block is a normal language Transformer. Its training end-to-end with the visual model can constrain the visual model and force it to take into account the language representation to build the visual embedding. This should result in more specific visual representations for sign language which has not yet been explored extensively in SLT. For this shared task, we use the Transformer for the language block with parameters shown in Table 1. This choice is motivated by Camgöz et al. (2020) which improved their previous results with LSTMs and GRUs (Camgoz et al., 2018) by more than 10 BLEU points. Furthermore, the Transformer makes the visual and language fusion more intuitive and easier for SLT, because it can process the whole sentence at the same time.

2.3 Loss and optimizer

In our experiments, we use a generalized loss. The general loss is considering both vision and text as a single model so the backpropagation starts from the last layer of the language part to the first layer of the visual one. We used the regular cross entropy loss from (Vaswani et al., 2017), with smoothing value = 0.1. Our optimizer has the following configuration: Adam with beta values =(0.9, 0.98), epsilon =1e-8, weight decay = 0.001.

Parameter	Value	Comments
Training corpus	FN+SRF	Remove sentences with >50 tokens
Batch size	10	Using few workers (<=5) on a single GPU
End of training criteria	PPL	Stop after 14 epochs without improvements
Language model	Transformer "base"	The number of encoder/decoder layers is 3 instead of 6
Visual model	3D ResNet (outsize= 2048, depth=50)	Additional custom module that converts the output size to our Sentence to Words Mapping (SWM)
SWM	32	Numbof the splits.
Scheduler	LambdaLR	Using warmup=4000
Max. output length	50	Maximum decoder output size
Gradient accumulation step	32	To get 320 sentences

Table 1: Main parameters used in training our primary submission DFKI-MLT.2.

Corpus	Sentences	Vocab	Min/Mean/Max/Std
SRF+FN	17192	26250	1/13.62/168/7.33
SRF	7056	14573	1/14.29/126/7.29
FN	10136	16723	1/13.15/168/7.32
SRF+FN dev	420	2003	2/13.98/44/6.95

Table 2: Text corpus statistics in tokens.

Corpus	Videos	Min	Mean	Max	Std
SRF	29	1492.6	1935.9	2106.2	106.3
FN	197	209.8	349.3	571.4	64.1
SRF+FN dev	420	0.6	5.84	19.86	3.42

Table 3: Number of videos and video statistics in seconds.

3 Setup and Experiments

3.1 Data description

For the submission, we use only the training and validation data given for the shared task and made up of FocusNews and SRF corpora, both parallel in Swiss German Sign Language and German text. SRF contains longer videos (approximately 30 minutes), FN contains more videos but shorter ones (approximately 5 minutes). The statistics of the German part of the corpus are summarized in Table 2 and the video statistics in Table 3.

3.2 Data preprocessing and batching

Since the input videos are long and contain more than one sentence (Table 3), we perform a subclipping step as preprocessing. By reading the subtitle files entries (*srt* in Figure 1(a)), we extract the time intervals and the corresponding sentences. We use ffmpeg to cut videos using these timestamps. We save the resulting subclips and add paths with subtitles (sentences) in one single annotation file.

We resize our input images to 224x224 pixels to leave a door open for pretraining approaches later. The batching is done using the Videodataset class (Wang et al., 2019). The depth is the number of frames in a video, it constitutes the third dimension in the 3D model. In our experiments, we initialize it to 100. To make sure that the language model keeps its original performance, we need to simulate a higher batch size. However, only a small number of videos can be placed in the same batch. We use gradient accumulation and update every 320 sentences for this purpose.

We do not do any preprocessing for the German textual data besides tokenization.

3.3 Experimental protocol

For the sake of reproducibility, we detail the setup for our primary submission in Table 1.

3.4 Evaluation

We use the same automatic metrics used by the shared task organisers in their preliminary automatic evaluation results (Müller et al., 2022). We use SacreBLEU (Post, 2018) to calculate BLEU[1] (Papineni et al., 2002) and chrF2++[2] (Popović, 2017). As semantic metric we use BLEURT[3] (Sellam et al., 2020).

4 Results and Analysis

Our best model according to BLEU is obtained with the largest 3D ResNet model and reaches 4.8 points on the development set, much higher than the performance of any system on the official test set. However, different metrics do not correlate, and chrF2++ and BLEURT —which correlate better with human judgments than BLEU— point towards a different model. Table 4 shows how per-

[1] BLEU|nrefs:1|bs:1000|seed:12345|case:mixed|eff:no|tok:13a|smooth:exp|version:2.2.0

[2] chrF2++|nrefs:1|bs:1000|seed:12345|case:mixed|eff:yes|nc:6|nw:2|space:no|version:2.2.0

[3] BLEURT v0.0.2 using checkpoint BLEURT-20

VisualModel	**BLEU**	**chrF2++**	**BLEURT**
ResNet50_3D	0.07 ± 0.02	8.07 ± 0.24	0.054 ± 0.003
ResNet34_3D	**4.82 ± 0.99**	8.28 ± 0.60	0.075 ± 0.007
ResNet10_3D	2.83 ± 1.41	**11.85 ± 1.32**	**0.100 ± 0.012**

Table 4: Results from different 3D ResNet scales on the development set.

Submission	**BLEU**			**chrF2++**			**BLEURT**		
	all	SRF	FN	all	SRF	FN	all	SRF	FN
UZH (Baseline)	0.12±0.06	0.09±0.03	0.19±0.11	4.7±0.4	4.5±0.5	5.0±0.7	0.102±0.006	0.095±0.006	0.110±0.009
DFKI-MLT.1	0.07±0.05	0.05±0.02	0.12±0.10	6.2±0.4	5.9±0.5	6.4±0.5	0.100±0.008	0.097±0.009	0.100±0.012
DFKI-MLT.2	0.11±0.06	0.08±0.03	0.17±0.13	6.3±0.4	6.4±0.6	6.1±0.6	0.083±0.008	0.074±0.008	0.091±0.013
DFKI-MLT.3	0.08±0.04	0.06±0.02	0.13±0.10	6.1±0.4	6.3±0.6	6.0±0.6	0.075±0.009	0.067±0.009	0.081±0.014
DFKI-MLT.4	0.02±0.01	0.02±0.01	0.04±0.02	3.9±0.2	3.7±0.3	4.1±0.3	0.066±0.004	0.063±0.004	0.070±0.008
DFKI-MLT.5	0.04±0.02	0.03±0.00	0.08±0.04	5.2±0.2	4.9±0.3	5.5±0.4	0.078±0.004	0.074±0.005	0.080±0.007

Table 5: Automatic evaluation of our 5 submissions and the shared task baseline on WMT-SLT test set (all), the SRF subset and the Focus News (FN) subset as provided by the organizers (Müller et al., 2022). DFKI-MLT.2 is our primary submission.

Hypothesis	**Reference**
Die -.	Die Diamantenschleiferei beschäftigt 63 Angestellte , davon 17 Behinderte , sowohl Rollstuhlfahrer als auch Gehörlose .
Der - .	Man arbeitet von 2004 bis 2009 ausbildungstechnisch mit dem Plussport Behindertensport Schweiz zusammen .
Und .	3 . Für die Sommer Deaf Olympics 2017 standen mehrere Städte zur Auswahl , nämlich Barcelona , Buenos Aires und Ankara .

Table 6: Sample outputs in the translation of the development set by the DFKI-MLT.3 system.

formance varies depending on the size of the 3D ResNet model. The smallest models seem to perform better across metrics and therefore we use ResNet10_3D in our submissions.

The low scores obtained with all our models correspond to a system that simply matches the most frequent words like "Die", "Der", "Und" as illustrated in Table 6. The rest of the generated sentence is a series of <UNK> tokens that are removed after decoding. We observe that training passes through some remarkable steps. It starts to output the most frequent words repeatedly, 1-grams, and as training advances the system starts to predict higher n-grams. In our experiments, the model stayed at the 1-gram stage.

We submitted 5 runs to the shared task, three of them using ResNet10_3D and the parameters are provided in Table 1. DFKI-MLT.1 was created with our main system using a checkpoint before the end of the training, DFKI-MLT.2 is the best checkpoint. We realized that both submissions had encoding issues and contain <UNK> tokens. We therefore sent a follow-up submission for DFKI-MLT.2, DFKI-MLT.3, containing the corrected format and without <UNK> tokens. As its translation quality was not even 0.5 BLEU points in the leaderboard, which may be less than a random walk from the vocabulary, we sent random walk results with repetitions (DFKI-MLT.4 and DFKI-MLT.5) to compare the performance.

A preliminary automatic evaluation has been made available by the organizers and it is shown in Table 5. Our final submission reached 0.11 ± 0.06 BLEU, 6.3 ± 0.4 chrF2++ and 0.083 ± 0.008 BLEURT, where confidence intervals are at 95% level. Results are therefore not statistically better than the baseline at 95% level. Interestingly, according to BLEURT, the random walk though the vocabulary is not significantly worse than the combination of 3D CNNs and Transformers.

In general, translation quality is always very bad, but results are slightly better for the FocusNew subset. FocusNews' input videos are shorter and this might imply a better alignment between videos and subtitles, improving the training. Some of our test outputs contain repetitions of (parts of) sentences from FocusNew dataset. Since this subcorpus is dominant in the final training (Table 2) the system is biased towards its vocabulary and this also explains the better performance in its subtest.

5 Conclusion

This paper presented an overview and some insights on spatio-temporal sign language representation which were used in the DFKI-MLT submission for the WMT-SLT 2022 shared task. To achieve our goal of building a fully end-to-end sign language model, we worked closely on the representation learning of visual features. Most of previous techniques for SLT simplify the feature representation by extracting spatial features and not benefiting from temporal features. This choice is motivated by the lack of temporal boundaries in sign language videos. To extract spatio-temporal features one can use 2D + 1D CNN approaches but this does not allow a fully end-to-end training as it still requires pretraining in another well-resourced task like object classification. In order to construct a specific representation model for SL and learn temporal modeling in a single model, we choose 3D CNNs and trained them from scratch simultaneously with the textual counterparts.

The translations produced by this architecture are very short and output only high frequency tokens; in few cases, full fluent and grammatical sentences are constructed but their meaning unrelated to the source. The generation of short sentences might be a limitation of our approach that builds a sentence representation with an output conversion method that does not split a sentence in subunits that can be weighted by the Transformer's attention mechanism to generate the output.

However, all the systems in this shared task's leaderaboard have translation scores close to zero. This shows the extreme difficulty of SLT and how bad current systems generalize to new data sets. We believe that system comparisons with such a bad translation quality do not allow to extract meaningful conclusions. In our future work, we investigate on different temporal modeling coupled with the 3D CNNs approach to further pursue the goal of developing a high-quality end-to-end system.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural sign language translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7784–7793.

Necati Cihan Camgöz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020. Sign language transformers: Joint end-to-end sign language recognition and translation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10020–10030. Computer Vision Foundation / IEEE.

Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. 2017. Learning spatio-temporal features with 3d residual networks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, pages 3154–3160.

Yuecong Min, Aiming Hao, Xiujuan Chai, and Xilin Chen. 2021. Visual alignment constraint for continuous sign language recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11542–11551.

Mathias Müller, Sarah Ebling, Avramidis Eleftherios, Alessia Battisti, Michèle Berger, Richard Bowden, Annelies Braffort, Necati Cihan Camgöz, Cristina España-Bonet, Roman Grundkiewicz, Zifan Jiang, Oscar Koller, Amit Moryossef, Regula Perrollaz, Sabine Reinhard, Annette Rios, Dimitar Shterionov, Sandra Sidler-Miserez, Katja Tissi, and Davy Van Landuyt. 2022. Findings of the WMT 2022 shared task on sign language translation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of* *the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. 2019. Temporal segment networks for action recognition in videos. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(11):2740–2755.

Experimental Machine Translation of the Swiss German Sign Language via 3D augmentation of body keypoints

Lorenz Hufe and Eleftherios Avramidis
German Research Center for Artificial Intelligence (DFKI Berlin)
Speech and Language Technology, Alt Moabit 91c, 10559 Berlin
lorenz.hufe@web.de, eleftherios.avramidis@dfki.de

Abstract

This paper describes the participation of DFKI-SLT at the Sign Language Translation Task of the Seventh Conference of Machine Translation (WMT22). The system focuses on the translation direction from the Swiss German Sign Language (DSGS) to written German. The original videos of the sign language were analyzed with computer vision models to provide 3D body keypoints. A deep-learning sequence-to-sequence model is trained on a parallel corpus of these body keypoints aligned to written German sentences. Geometric data augmentation occurs during the training process. The body keypoints are augmented by artificial rotation in the three dimensional space. The 3D-transformation is calculated with different angles on every batch of the training process.

1 Introduction

Despite the enormous progress of the Machine Translation (MT) of spoken (and written) languages, the MT of sign languages is in a very early stage (Yin et al., 2021; De Coster et al., 2022). Two major challenges are (a) the multimodal and multilateral nature of the sign languages and (b) the lack of data. On the one side, the multilateral and multimodal nature of the sign languages requires deep-learning topologies that differ substantially from the ones used in text-based MT. On the other side, the lack of data makes difficult the utilization of end-to-end deep learning algorithms, which usually require vast amounts of data. As a result, deep-learning experiments have been executed for very few sign languages (e.g. German Sign Language, DGS; American Sign Language, ASL) and narrow domains (e.g. weather forecasts), leaving open questions on the generalization of the methods to other sign languages and broader domains.

This year's Sign Language Translation (SLT) Task of the Seventh Conference of Machine Translation (WMT22) is contributing significant to this direction, by adding a new language pair (Swiss German Sign Language - DGSG - to German) and allowing extensive experimentation from several participants on the same dataset.

Our system uses computer vision models to analyze the sign language videos into body keypoints and uses these keypoints as the source-side input of the neural MT transformer, allowing to perform data augmentation via geometrical augmentations. Despite the difficulty of this shared task and the low results obtained, we publish this paper as a technical report, with the hope that it can contribute to the further research of this direction.

The rest of the paper is organized as following. Section 2 positions our contribution amidst related work. Section 3 describes the methods for training the system and Section 4 the technical set-up of the experiment. Section 5 provides and discusses some results, while Section 6 gives some conclusion and ideas for further work.

2 Related Work

Latest work on MT of sign languages has shown significant improvements using deep learning methods from the fields of computer vision and MT. State of the art work (Camgöz et al., 2018; Yin and Read, 2020; Camgöz et al., 2020; Zhou et al., 2022) employees transformers, which are given frame embeddings extracted from the videos of the signers.

Contrary to the use of pixel-based frame embeddings, Nunnari et al. (2021) suggests to use body keypoints from the hands, the skeleton and the face as input to the transformers. This requires to split the translation pipeline into a first phase, recognizing 3D keypoints from videos, and has the advantage that they can be augmented by applying transformation techniques. Our paper presents an implementation of that idea, applied to the case of DSGS.

The use of body keypoints has been considered

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 983 - 988
December 7-8, 2022 ©2022 Association for Computational Linguistics

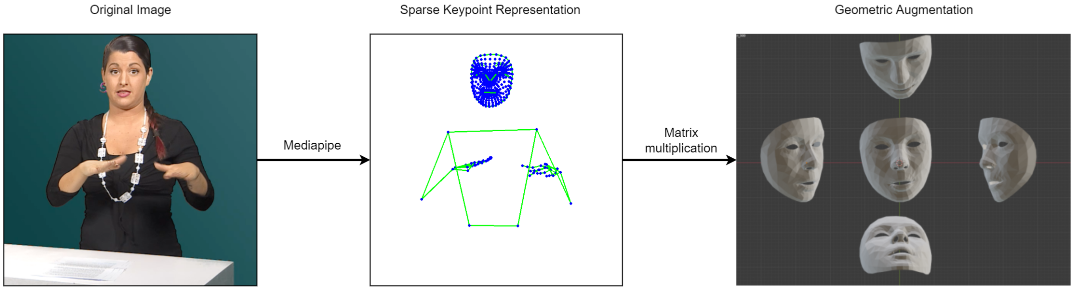


Figure 1: Mediapipe is used to extract sparse keypoint representations of the signer. The nature of the resulting 3D data allows for rotation, translation and shearing using matrix multiplication at virtually no cost.

by Gan et al. (2021), where skeleton pose information is processed together with the video frame input. Ko et al. (2018, 2019) use 2D coordinates of body keypoints to train the neural MT systems, but contrary to our work, they do not perform any geometrical transformations to the keypoints. Moryossef et al. (2021) analyze the applicability of the pose estimation systems to sign language recognition by evaluating the failure cases of the recognition models.

3 Method

Our system consists of three modules. The first module converts images of the signer into intermediate keypoint representations. The second module employs data augmentation to increase sample efficiency and decrease the effect of spurious feature correlations. Spurious data correlation in high dimensional spaces can lead to Clever Hans effects (Kauffmann et al.). The last module is the trainable transformer that translates from keypoint representation to German text, while interacting with the augmentation module.

3.1 Keypoint extraction

There are multiple reasons to believe that keypoint representations could prove beneficial in SLT. Only few and small datasets are available for SLT. That is because firstly there are only few known data sources for SL. Secondly the data transcription for SL needs expert knowledge which is costly and hard to find. Thirdly SL data inherently needs video footage of signing human, which makes anonymisation near impossible thus leads to privacy problems when detecting new potential data sources.

A end-to-end SLT pipeline needs to make sense of the movement of the human signer and translate these motions into written language. Practically speaking this means the pipeline internally needs to learn two tasks on limited data. However only the translation task depends on the costly and limited SLT datasets, while the task of detecting the motion could be eased by employing pose estimation which is not specific to SLT and therefore is more explored and cheaper in terms of data acquisition.

The extraction of the keypoints was done by using the computer vision models of MediaPipe Holistic (Grishchenko and Bazarevsky, 2020) which combines three pre-trained computer vision pipelines that detect the hand keypoints (MediaPipe Hands; Zhang et al., 2020), the keypoints of the body pose (BlazePose; Bazarevsky et al., 2020), and a keypoint mesh for the face (BlazeFace; Bazarevsky et al., 2019).

When data points were missing, the values were substituted by zero values.

3.2 Geometrical transformation

The geometrical transformation is applied during the training process of the transformer model. For every iteration of the training process, the 3D keypoints are given to the geometrical transformation module. This returns the co-ordinates of the original keypoint mesh after being rotated. The 3D keypoints get rotated around the x, y and z axis by some angle R_x, R_y and R_z respectively, using rotation matrices. First, the rotation around the x-axis takes place, followed by y and then the z axis.

The rotation angle is drawn at random at every training iteration, such that every batch is rotated to a different setting. The angle of the rotation is limited to a particular range, which makes sense for the particular axis. R_x is drawn from [-60°, +60°] while R_y and R_z are drawn from [-10°, +10°].

3.3 Sequence-to-sequence model

The sequence-to-sequence model is based on a NMT transformer model similar to (Camgöz et al., 2018). We provide the network the keypoint representation, by concatenating all mediapipe keypoints and then flattening them into a 708 dimensional vector. The target language is the Swiss German text.

4 Experiment setup

The experiment took place using only the corpora FocusNews, as provided by the shared task organizers, including keypoints precomputed with MediaPipe. Due to time restrictions, the SRF corpus was not used, since it did not provide any keypoints. The training set had 10,136 sentences, the validation set 420 sentences and the test set 488 sentences. Due to problems with the keypoint-subtitle alignment only 393 of the 420 sentences of the validation set were used.

For training the model we modified the NMT toolkit JoeyNMT[1] (Kreutzer et al., 2019), extending the SLT branch created by Camgöz et al. (2020). We followed the text pre-processing of the previous implementation, which included text lowercasing. The geometrical transformations were done with array computations using NumPy (Harris et al., 2020). The automatic evaluation metrics were computed using SacreBLEU (Post, 2018).

In order to optimize the system we ran several experimental rounds. The training parameters for all rounds can be seen in Table 2. The experimental rounds were run by modifying the following parameters:

- **max. rotation**: The maximum angle for the random rotation that took place for every iteration. A max. rotation of 10° here means that for every iteration batch, a random degree value within [-10°, +10°] was drawn.
- **patience**: The learning rate scheduler stops when no significant progress is measured with the evaluation metric, after a number of epochs. This parameter defines how patient the scheduler is in that regards.
- **LR scheduler metric**: The metric used for measuring the progress on the validation set.
- **layers**: The number of layers for the encoder and the decoder of the transformer.

5 Results

As part of our parameter we ran 5 experimental rounds which are shown in Table 1. Due to time limitations it was not possible to experiment with the full spectrum of parameters, including ablation tests which would indicate the contribution of possible parameter values. Even in that case, the very low metric scores would not lead to more significant conclusions.

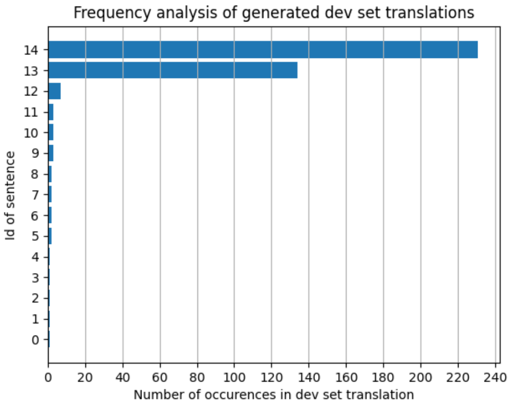


Figure 2: Overview over the translation sentence frequencies over the dev set

From the first experiments it was obvious that the use of BLEU-4 as a validation metric could not contribute to the optimization, because its values are always zero and also the training time was very short. For this reason we chose ChrF as validation metric for our last two experiments. Increasing the patience deemed necessary, so that the training mechanism can get enough random samples from the augmentation process. For our best iteration we experimented with both 3 and 4 layers, resulting into slightly better performance with the 4 layer setting.

In overall, the results of our experiments, as measured by automatic metrics, showed very low performance. No version of our pipeline could achieve non-zero BLEU-4 score on the provided development set, meaning that no n-gram of order 4 was correctly matched between the hypothesis and the reference. The experiments measured with BLEU-3 and ChrF indicate as better run the configuration with 60 degrees rotation range at the X axis, 10 degrees on the other axes, and a patience of 500. When analyzing the output on the validation set we found that for the 393 different sentences of the validation set, only 15 different translations were repeatedly produced as highlighted in figure 2 and listed in Appendix A. The two most common translations make up for 92% of the cases. This behaviour suggests that the model learned two main

[1] Our code is available at https://github.com/DFKI-SignLanguage/slt under Apache 2.0 License

max rotation +/- (°)			LR scheduler			scores			
x	y	z	patience	metric	layers	BLEU-3	BLEU-4	ChrF	runtime (h)
10	10	10	25	BLEU	4	0,28	0,00	15,36	00:21
10	10	10	50	BLEU	4	0,28	0,00	15,36	00:31
60	10	10	50	BLEU	4	0,00	0,00	17,58	00:24
60	10	10	500	ChrF	3	0,310	0,00	16,08	07:44
60	10	10	500	ChrF	4	0,314	0,00	16,43	04:14

Table 1: Overview over the results on the validation set when employing different settings.

parameter	value
feature size	708
max sentence len.	400
dropout	0,1
FF size	2048
heads	8
embeddings dim.	512
hidden size	512
optimizer	adam
batch size	32
random seed	42
weight decay	0,001
learning rate	0,001
validation freq.	100
beam size	1
beam alpha	-1
translation max len.	30

Table 2: Training parameters

prototype translations and is not sensitive to the input when translating.

6 Conclusion and Further Work

Due to the poor results, very little can be concluded about the effect of the proposed geometric augmentation strategy. As suggested by the preliminary results of the shared task (Müller et al., 2022) no group was able to achieve good results on the task. Unfortunately, due to the strict workshop timeline we could not perform further experiments to empirically prove the causes of this low performance. We are planning to do this in future work, including an ablation study of the different modules and a comparison with the state-of-the-art on other datasets. Further research should be invested in exploring the possible use cases for geometric data augmentation in MT of SL.

Acknowledgements

This work has been funded by the German Ministry of Education and Research through the project SocialWear (01IW20002). Many thanks to our colleagues Cristina España i Bonet, Yasser Hamidullah and Fabrizio Nunnari for providing ideas and support for this experiment.

References

Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, and Matthias Grundmann. 2020. BlazePose: On-device Real-time Body Pose tracking. *CoRR*, abs/2006.1.

Valentin Bazarevsky, Yury Kartynnik, Andrey Vakunov, Karthik Raveendran, and Matthias Grundmann. 2019. BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs. *CoRR*, abs/1907.0.

Necati Cihan Camgöz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural Sign Language Translation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 7784–7793, San Francisco, CA, USA. IEEE.

Necati Cihan Camgöz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020. Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 10020–10030. Institute of Electrical and Electronics Engineers (IEEE).

Mathieu De Coster, Dimitar Shterionov, Mieke Van Herreweghe, and Joni Dambre. 2022. Machine Translation from Signed to Spoken Languages: State of the Art and Challenges.

Shiwei Gan, Yafeng Yin, Zhiwei Jiang, Lei Xie, and Sanglu Lu. 2021. Skeleton-Aware Neural Sign Language Translation. *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4353–4361.

Ivan Grishchenko and Valentin Bazarevsky. 2020. MediaPipe Holistic — Simultaneous Face, Hand and Pose Prediction, on Device.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

Jacob Kauffmann, Lukas Ruff, Grégoire Montavon, and Klaus-Robert Müller. The clever hans effect in anomaly detection.

Sang-Ki Ko, Chang Jo Kim, Hyedong Jung, and Choongsang Cho. 2019. Neural sign language translation based on human keypoint estimation. *Applied Sciences*, 9(13):2683.

Sang-Ki Ko, Jae Gi Son, and Hyedong Jung. 2018. Sign language recognition with recurrent neural network using human keypoint detection. In *Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems*, RACS '18, page 326–328, New York, NY, USA. Association for Computing Machinery.

Julia Kreutzer, Jasmijn Bastings, and Stefan Riezler. 2019. Joey NMT: A minimalist NMT toolkit for novices. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 109–114, Hong Kong, China. Association for Computational Linguistics.

Amit Moryossef, Ioannis Tsochantaridis, Joe DInn, Necati Cihan Camgoz, Richard Bowden, Tao Jiang, Annette Rios, Mathias Muller, and Sarah Ebling. 2021. Evaluating the Immediate Applicability of Pose Estimation for Sign Language Recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 3429–3435, Nashville, TN, USA. IEEE Computer Society.

Mathias Müller, Sarah Ebling, Eleftherios Avramidis, Alessia Battisti, Michèle Berger, Richard Bowden, Annelies Braffort, Necati Cihan Camgöz, Cristina España-Bonet, Roman Grundkiewicz, Zifan Jiang, Oscar Koller, Amit Moryossef, Regua Perrollaz, Sabine Reinhard, Annette Rios, Dimitar Shterionov, Sandra Sidler-Miserez, Katja Tissi, and Davy Van Landuyt. 2022. Findings of the WMT 2022 shared task on sign language translation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Fabrizio Nunnari, Cristina España-Bonet, and Eleftherios Avramidis. 2021. A data augmentation approach for sign-language-to-text translation in-the-wild. In *Proceedings of the 3rd Conference on Language, Data and Knowledge*, volume 93 of *OpenAccess Series in Informatics, OASIcs*, Zaragoza, Spain. Dagstuhl publishing.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Kayo Yin, Amit Moryossef, Julie Hochgesang, Yoav Goldberg, and Malihe Alikhani. 2021. Including signed languages in natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7347–7360, Online. Association for Computational Linguistics.

Kayo Yin and Jesse Read. 2020. Better sign language translation with STMC-transformer. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5975–5989, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. 2020. MediaPipe Hands: On-device Real-time Hand Tracking. *CoRR*, abs/2006.1.

Hao Zhou, Wengang Zhou, Yun Zhou, and Houqiang Li. 2022. Spatial-Temporal Multi-Cue Network for Sign Language Recognition and Translation. *IEEE Transactions on Multimedia*, 24:768–779.

Appendix

A Translations

0. **Empty**
1. die eltern sind sehr engagiert und kämpfen für die pille verbieten.
2. das ziel der konferenz sind vorträge von swisscom zu zeigen, dass diese kinder noch nicht gebärdensprache.
3. das ziel der konferenz sind vorträge von swisscom zu zeigen, dass diese kinder noch nicht zugänglich.
4. das ziel der konferenz sind vorträge von swisscom zu zeigen, dass sie sich nicht mit einer behinderung einsetzen.
5. die postverteilungs-firma
6. bis zum nächsten mal.
7. die eltern sind sehr engagiert und kämpfen für die gebärdensprache, ihre tochter hat.
8. das ziel der swisscom ist eine optimale beratung und einen guten service anzubieten.
9. die gehörlosen kinder freuten sich sehr, da sie alles verstanden und somit integriert geschult integriert geschult werden schulen sollen.
10. die eltern sind gehörlos.
11. die eltern sind sehr engagiert und kämpfen für die hochschule.
12. die forscher meinen, dass kinder mit cochlea-implantate über eine genauso gute lebensqualität wie hörende kinder verfügen, ohne psychosoziale folgen.

13. die eltern sind sehr engagiert und kämpfen für die gebärdensprache, ihre kultur und ihre rechte.
14. die voraussetzungen für diese stelle sind ein kürzlich abgeschlossenes hochschulstudium sowie die bereitschaft, arbeiten im sinne der gleichstellung zu schreiben.

TTIC's WMT-SLT 22 Sign Language Translation System

Bowen Shi
TTI-Chicago
bshi@ttic.edu

Diane Brentari
Univeristy of Chicago
dbrentari@uchicago.edu

Greg Shakhnarovich
TTI-Chicago
greg@ttic.edu

Karen Livescu
TTI-Chicago
klivescu@ttic.edu

Abstract

We describe TTIC's model submission to WMT-SLT 2022 task (Müller et al., 2022) on sign language translation (Swiss-German Sign Language (DSGS) → German). Our model consists of an I3D backbone for image encoding and a Transformer-based encoder-decoder model for sequence modeling. The I3D is pretrained with isolated sign recognition using the WLASL dataset. The model is based on RGB images alone and does not rely on the pre-extracted human pose. We explore a few different strategies for model training in this paper. Our system achieves 0.3 BLEU score and 0.195 Chrf score on the official test set.

1 Introduction

Sign language, a full-fledged natural language that conveys meaning through gestures, is the primary chief of communication among Deaf people. Sign language translation is a task for automatically translating sign languages into written languages. Due to its widespread potential applications, it has recently received growing research interest (Camgoz et al., 2018, 2021).

Existing methods for sign language translation are primarily based on gloss, a transliteration system annotating sign language with symbols from written language. Utilizing gloss usually boosts the performance of current translation systems by a large margin. In the widely used German sign language translation benchmark Phoenix14T (Camgoz et al., 2018), state-of-the-art gloss-based models (Chen et al., 2022) are roughly 15 points better (in Bleu-4) than gloss-free models (Camgoz et al., 2018). However, gloss is more expensive to annotate than written language translation. There have been relatively few amounts of studies for gloss-free sign language translation. Specifically, Orbay and Akarun (2020); Shi et al. (2022) utilize local visual features (e.g., hands) to enhance the translation performance. Those systems require domain-specific training data (e.g., labeled handshape data used in Orbay and Akarun (2020)), which is not always accessible for the target sign language. The fusion of visual features at different scales also increases the complexity of the modeling pipeline.

In this paper, we study a simple model for sign language translation between DSGS and German in a gloss-free setting. Our model uses a 3D convolutional network for visual feature extraction and a Transformer-based encoder-decoder for sequence modeling. It is built on raw RGB images rather than pose keypoints, thus avoiding potential mistakes from pose estimation and remaining fast in inference. We further study the impact of hyperparameters and different pretraining strategies on translation quality. Without ensembling, our model achieves 0.3 Bleu score and 0.195 Chrf score on the official test set.

2 Method

In this section, we describe our method for sign language translation. Our model consists of an Inflated 3D ConvNet (I3D) (Carreira and Zisserman, 2017) for visual encoding and a Transformer-based encoder-decoder model (Vaswani et al., 2017) for sequence modeling, which are described respectively below.

I3D I3D (Carreira and Zisserman, 2017) is a 3D convolutional neural network proposed in action recognition. I3D has previously been explored in sign language processing (Albanie et al., 2020; Li et al., 2020; Vaezi Joze and Koller, 2019) and achieved competitive performance in isolated sign recognition (Li et al., 2020). More formally, given a sequence of image frames $\mathbf{I}_{1:T}$, the I3D model $\mathbf{M}^v$ encodes them into a sequence of visual features $\mathbf{f}_{1:T'}$: $\mathbf{f}_{1:T'} = \mathbf{M}^v(\mathbf{I}_{1:T})$, where T and T' respectively denote the length of video and visual feature sequence. Note due to the temporal stride in convolutional kernels of I3D, T' is not equal to T and is usually several factors smaller.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 989 - 993
December 7-8, 2022 ©2022 Association for Computational Linguistics

To encourage the visual encoder $\mathbf{M}^v$ to capture more signing-related visual cues (e.g., arm movement, handshape, and so on), we pretrain the I3D model with isolated sign recognition on WLASL, a large-scale dataset consisting of isolated American sign language (ASL) signs. Though ASL and DSGS are two different sign languages, visual features regarding body movement are shared. Empirically, we observed considerable gains in isolated sign pretraining. For computational efficiency, the pretrained I3D network $\mathbf{M}^v$ is frozen in translation model training.

Transformer-based encoder-decoder We employ a Transformer-based encoder-decoder (Vaswani et al., 2017) model $\mathbf{M}^{(s)}$ to decode visual feature $\mathbf{f}^{(v)}_{1:T'}$ into text $w_{1:N}$: $w_{1:N} = \mathbf{M}^{(s)}(\mathbf{f}^v_{1:T})$. $\mathbf{M}^s$ is a standard sequence-to-sequence model widely used in machine translation (Vaswani et al., 2017; Barrault et al., 2020; Akhbardeh et al., 2021). Thus we only briefly review it here and a more detailed description can be found in Vaswani et al. (2017). Our sequence-to-sequence model $\mathbf{M}^s$ includes a Transformer encoder and Transformer decoder, which are joined via attention. Specifically, the Transformer encoder transforms the visual features $\mathbf{f}^v_{1:T}$ into $\mathbf{e}_{1:T}$ by injecting temporal information based on self-attention and positional embedding. The Transformer decoder generates token sequence $w_{1:N}$ in an auto-regressive manner while attending to the encoder output $\mathbf{e}_{1:T}$ through the attention mechanism.

Training loss We use cross-entropy loss for model training. More formally, given the translation pair $(\mathbf{I}_{1:T}, \hat{w}_{1:N})$, suppose the model outputs probability vector $\mathbf{p}(\cdot|\mathbf{I}_{1:T}, \hat{w}_{1:n-1})$ at decoder step n. The loss is then computed as

$$l = -\sum_{n=1}^{N} \log p(\hat{w}_n|\mathbf{I}_{1:T}, \hat{w}_{1:n-1}) \qquad (1)$$

Inference At test time, we use beam search for decoding image sequence $\mathbf{I}_{1:T}$. The beam width and length penalty are hyperparameters tuned using the development set.

3 Experimental Setup

Data We use FocusNews and SRF data to train our translation model. Both FocusNews and SRF consist of DSGS-German pairs, which include 19 and 16 hours (10,136 and 7,071 sequences) of DSGS videos, respectively. The two datasets differ in multiple aspects. For example, FocusNews are live signing from teleprompters by deaf signers based on news from 2008 to 2014, whereas SRF dataset contains news videos from 2020 to 2021 which is interpreted into DSGS by hearing interpreters. Both datasets are incorporated into training. Note that frame rates in videos of FocusNews and SRF differ, we feed the raw videos in FocusNews and SRF without frame rate conversion. To pretrain the visual encoder, we use WLASL (Li et al., 2020), a large-scale isolated sign dataset including $\sim 21k$ pairs of American sign language video clips and English words.

Training We use sentencepiece unigram tokenizer (Kudo, 2018) to tokenize the German translation. The number of subword units is tuned to 18,000 We use a 2-layer Transformer with 512 hidden dimensions and 2048 hidden dimensions for both encoder and decoder. A dropout layer with a zeroing probability of 0.1 is added between the self-attention layer and the feedforward network. The model is trained with Adam (Kingma and Ba, 2015) for 18K steps at a batch size of 32. The learning rate is linearly increased to 0.0008 for 2K steps and decayed to 0 in the remaining steps. The visual backbone I3D is pretrained on WLASL and frozen during translation model training. During isolated sign training, we initialize I3D from a model trained on the action recognition dataset Kinetics (Carreira and Zisserman, 2017). We use SGD with a 0.9 momentum value to train the model for 50 epochs at a batch size of 4. The initial learning rate is 0.001 and is halved if accuracy on the validation set does not increase for 3 epochs. Before feeding into I3D, each isolated sign video is truncated to a 64-frame clip, which is padded with all-zero frames if the length is shorter than 64. Each image frame is resized to 240 × 240. It is randomly cropped to 224 × 224 and horizontally flipped at a probability of 0.5 in training. At test time, we only center cropping to every image frame.

Evaluation We evaluate the system using BLEU-{1,2,3,4} (Lin, 2004) and ROUGE (Lin, 2004) scores.

4 Experimental Results

In this section, we report results and conduct some analyses of the translation model on the development set.

Hypothesis	Reference	Bleu-4
das der stand der dinge im moment. gibt es eine grosse aufsp. (that's the state of things at the moment. is there a big sp.)	das der stand der dinge im moment. (that's the state of things at the moment.)	51.56
mit live-untertiteln von swiss txt guten abend, meine damen und herren, willkommen zur "tagesschau" (with live subtitles from swiss txt good evening, ladies and gentlemen, welcome to the "tagesschau")	guten abend, meine damen und herren, willkommen zur "tagesschau". (good evening, ladies and gentlemen, welcome to the "tagesschau".)	51.42
die armee muss ihre arbeit nicht mehr einmal. (the army doesn't even have to do its job anymore.)	doch die bevölkerung macht nicht mit. (but the population does not participate.)	0.00
die französischen roben programm speziell für gehörlose. (the french robes program especially for the deaf.)	bei auf der webseite des sportverbandes können detailliertere informationen nachgelesen werden. (more detailed information can be found on the website of the sports association.)	0.00

Table 1: Qualitative examples produced by our translation system. The sentence within () is the corresponding English translation.

4.1 Main Results

Table 2 shows the performance of our model on the development set compared to the Sockeye baselines reported from the official repo (Müller et al., 2022). Our model outperforms Sockeye baselines, which are models based on the pre-extracted human pose. However, the overall values in different metrics are very low. We further show translation examples produced by our model (see Table 1). We noticed the phrases that are translated correctly by our model are usually duplicate phrases frequently appearing in training (e.g., willkommen zur "tagesschau"). For most of the sentences, the model is unable to capture its meaning generally though many predictions are grammatically correct. Such observation shows that large-vocabulary sign language translation is very challenging.

	Train Data	Rouge	B1	B2	B3	B4
Sockeye (Müller et al., 2022)	FN	-	-	-	-	0.21
Sockeye (Müller et al., 2022)	Srf	-	-	-	-	0.59
Sockeye (Müller et al., 2022)	FN,Srf	-	-	-	-	0.15
Ours	FN,Srf	7.92	8.36	2.92	1.55	**1.02**

Table 2: Performance of our model on development set. The Sockeye baselines are from the official repo (Müller et al., 2022). FN: FocusNews

4.2 Hyperparameter Tuning

Among the set of hyperparameters, we find that the following two hyperparameters have the most significant effect on translation performance: learning rate and the number of layers. We detail their impact on model performance below. Other hyperparameters (e.g., dropout, learning rate schedule) are also tuned in our experiments. However, their impact is relatively negligible and thus not detailed in the paper.

Learning rate We tuned the learning rate among $\{0.001, 0.002, 0.004, 0.008, 0.016\}$. As is shown in Table 3, increasing the learning rate consistently improves the model performance across all the metrics. The benefit plateaus around 0.008, which is the optimal value among the set of values we consider.

Number of layers We further tuned the number of Transformer layers (see Table 4). We keep the number of encoder and decoder layers the same and set the hidden/feedforward dimension to 512/2048 in the corresponding experiments of Table 4. Increasing number of transformer layers degrades the performance. This is probably because the 3D convolutional kernels of I3D capture some temporal relations in the video, which reduces the reliance of

LR	Rouge	B1	B2	B3	B4
0.001	7.82	8.38	2.51	1.21	0.76
0.002	6.85	7.25	2.22	1.05	0.69
0.004	6.86	6.08	2.22	1.18	0.82
0.008	**7.92**	**8.36**	**2.92**	**1.55**	**1.02**
0.016	7.54	6.11	2.27	1.35	1.01

Table 3: Impact of learning rate on translation performance

the whole model on Transformer modules to capture sequential information. Furthermore, larger models (i.e., more layers) usually require more training data. The total amount of sign language videos (35 hours) is probably insufficient to train a deep transformer encoder-decoder.

# Layer	Rouge	B1	B2	B3	B4
2	**7.92**	**8.36**	**2.92**	**1.55**	**1.02**
4	7.10	7.01	2.31	1.21	0.82
6	6.17	7.32	1.55	0.52	0.24

Table 4: Impact of Transformer layers on translation performance

4.3 Effect of I3D pretraining

The I3D backbone is pretrained on WLASL. Here we compare three options of I3D pretraining: WLASL, BSL-1K, and Kinetics-400. BSL-1K is a coarticulated sign dataset of 1064 British sign language (BSL) signs (273K video clips in total), collected from BBC videos interpreted into BSL. Kinetics (Carreira and Zisserman, 2017) is the action recognition dataset with 650K videos from 400 human action categories. As is shown in Table 5, pretraining with sign-language specific datasets (WLASL, BSL-1K) consistently outperforms pretraining with general human action videos (Kinetics). This is expected as signing-related visual cues (e.g., handshapes), essential for sign language translation, are better captured in isolated sign datasets. Pretraining with WLASL achieves better results than BSL-1K. Though BSL-1K contains an overall larger number of video clips than WLASL (21K vs. 273K), it has fewer unique signs (1064 vs. 2000). This probably suggests that a sign language corpus with more signing categories will be more beneficial to sign language translation compared to its counterpart with fewer signs.

PT Data	Rouge	B1	B2	B3	B4
WLASL	**7.92**	**8.36**	**2.92**	**1.55**	**1.02**
BSL-1K	6.88	6.19	1.86	0.84	0.69
Kinetics	5.05	4.18	1.02	0.65	0.41

Table 5: Impact of Transformer layers on translation performance

5 Conclusion

This paper describes TTIC's DSGS-German translation system submitted to the WMT-SLT 2022 challenge. Our model consists of an I3D model for visual feature extraction and a Transformer-based encoder-decoder for sequence modeling. The system is based on RGB images alone and remains conceptually simple. Our experiments show that pretraining the visual frontend with isolated sign recognition helps achieve better translation performance. However, the overall translation quality is still in a very low regime. Our future work includes combining pose and RGB-based models for sign language translation.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondrej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Samuel Albanie, Gül Varol, Liliane Momeni, Triantafyllos Afouras, Joon Son Chung, Neil Fox, and Andrew Zisserman. 2020. BSL-1K: Scaling up co-articulated sign language recognition using mouthing cues. In *ECCV*.

Loïc Barrault, Magdalena Biesialska, Ondrej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubesić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos

Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

N. C. Camgoz, B. Saunders, G. Rochette, M. Giovanelli, G. Inches, R. Nachtrab-Ribback, and R. Bowden. 2021. Content4all open research sign language translation datasets. *ArXiv*, abs/2105.02351.

Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural sign language translation. In *CVPR*, pages 7784–7793.

João Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. *CVPR*, pages 4724–4733.

Yutong Chen, Fangyun Wei, Xiao Sun, Zhirong Wu, and Stephen Lin. 2022. A simple multi-modality transfer learning baseline for sign language translation. In *CVPR*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *ACL*.

Dongxu Li, Cristian Rodriguez-Opazo, Xin Yu, and Hongdong Li. 2020. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *WACV*, pages 1448–1458.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *ACL*.

Mathias Müller, Annette Rios, and Amit Moryossef. 2022. Sockeye baseline models for sign language translation. `https://github.com/bricksdont/sign-sockeye-baselines`.

Alptekin Orbay and Lale Akarun. 2020. Neural sign language translation by learning tokenization. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 222–228.

Bowen Shi, Diane Brentari, Greg Shakhnarovich, and Karen Livescu. 2022. Open-domain sign language translation learned from online video. *ArXiv*, abs/2205.12870.

Hamid Vaezi Joze and Oscar Koller. 2019. Ms-asl: A large-scale data set and benchmark for understanding american sign language. In *The British Machine Vision Conference (BMVC)*.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Tackling Low-Resourced Sign Language Translation: UPC at WMT-SLT 22

Laia Tarrés[1,2*] and **Gerard I. Gállego**[1,*] and **Xavier Giró-i-Nieto**[1] and **Jordi Torres**[2]
[1]*Universitat Politècnica de Catalunya, Spain* [2]*Barcelona Supercomputing Center, Spain*

Abstract

This paper describes the system developed at the Universitat Politècnica de Catalunya for the Workshop on Machine Translation 2022 Sign Language Translation Task, in particular, for the sign-to-text direction. We use a Transformer model implemented with the Fairseq modeling toolkit. We have experimented with the vocabulary size, data augmentation techniques and pretraining the model with the PHOENIX-14T dataset. Our system obtains 0.50 BLEU score for the test set, improving the organizers' baseline by 0.38 BLEU. We remark the poor results for both the baseline and our system, and thus, the unreliability of our findings.

1 Introduction

The submission of the Universitat Politècnica de Catalunya (UPC) to the WMT22 Sign Language Translation (SLT) Task experimented with vocabulary size, data augmentation techniques and a pretrained system with the PHOENIX-14T dataset (Camgoz et al., 2018). Up to the author's knowledge, our implementation is the first to build on Fairseq, a popular modeling toolkit by Meta AI (Ott et al., 2019).

SLT is a highly complex task because sign language understanding requires a very precise estimation of the signer pose, especially, of its hands. In addition, sign languages have grammatical structures different from spoken languages, which prevents an easy knowledge transfer. Sign languages are represented in continuous and high-dimensional spaces, while the transcribed version of spoken languages are represented by discrete tokens of well-defined vocabularies. Moreover, the few available sign language datasets can be considered very low-resourced, as they typically contain less than one hundred thousand sentences (Goyal et al., 2022). In particular, the total number of sentences from the two datasets provided in WMT-SLT22 is 17k.

We focus on the sign language translation task of WMT-SLT22 which requires participants to predict the translation in spoken language (written) from a sign language video. Specifically, it consists of translation from Swiss German Sign Language (DSGS) videos to German (DE) text.

The organizers of WMT 2022 SLT track propose a Transformer baseline (Müller et al., 2022) that achieves a very low BLEU (Papineni et al., 2002) score,[1] which indicates a very poor translation quality. The training data provided consists of two datasets: FocusNews (Müller et al., 2022b) and SRF (Müller et al., 2022a). The first one contains 197 episodes in DSGS that have an average length of 5 minutes, which amount for a total duration of 19 hours. The second dataset contains 29 videos from live sign language interpretation, that have an average of 30 minutes length, and totals a duration of 16 hours. The paired subtitles are given in Standard German from Switzerland, which is a dialect of German.

The organizers provide keypoints extracted with two off-the-shelf human pose estimators: OpenPose (Cao et al., 2019) and MediaPipe (Lugaresi et al., 2019).

2 Baseline system

For the sake of self-containment, we briefly define and discuss the baseline model proposed by the organizers for this task (Müller et al., 2022). This solution is built on sockeye (Hieber et al., 2017), with results for the official development (dev) partition.

Regarding the model implementation, they use a Transformer encoder-decoder, with a symmetrical number of layers for the encoder and decoder. The architecture has 6 layers, 8 heads, 2048 neurons in

[*] Equal contribution

[1] For simplicity, we refer to BLEU-4 score as BLEU score.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 994 - 1000
December 7-8, 2022 ©2022 Association for Computational Linguistics

the feed forward network layers, and an embedding dimension of 512.

The input data are 2D OpenPose keypoints, concatenating the hands and body landmarks. This results in an array $X = x_0, \ldots, x_T$ where $x_t \in \mathbb{R}^{2K}$, K is the number of keypoints selected, and T the number of video frames.

Three baseline results were provided by the WMT-SLT22 organizers, shown in Table 1. The first two scores correspond to the models trained individually with each of the two benchmark datasets. The last result is from a model trained on both datasets, which obtains the lowest performance. We hypothesize this might be due to a domain shift between the FocusNews and SRF datasets. These BLEU scores are extremely low, as already noted by Müller et al. (2022). As a comparison, the SLT state-of-the-art BLEU score for PHOENIX-14T dataset (Camgoz et al., 2018) is 25.59 (Voskou et al., 2021) and for the How2Sign dataset (Duarte et al., 2021) is 1.25 (Duarte et al., 2022).

Train dataset	BLEU
FocusNews	0.216
SRF	**0.589**
FocusNews + SRF	0.157

Table 1: Results provided by the organizers for the official dev partition, which contains FocusNews + SRF samples. Scores on test partition, which also contains FocusNews + SRF samples, were not released by the organizers.

3 Method

Our submission also adopts a Transformer architecture, which we implement with the Fairseq sequence modeling toolkit (Ott et al., 2019). Up to the author's knowledge, this is the first time that Fairseq is used for sign language video understanding. We publish our source code [2], offering the SLT community a novel tool widely used in machine translation for spoken languages. On top of this, we experiment with the vocabulary size, data augmentation techniques and pretraining the system with PHOENIX-14T dataset. Details are described in this section.

3.1 Preprocessing steps

The WMT-SLT22 organizers provide both OpenPose and MediaPipe keypoints from the body pose estimators. We choose MediaPipe poses because they provide 3D coordinates (x, y, z) normalized between $[0, 1]$. Moreover, based on our experience, OpenPose is more prone to errors, like detecting several people in videos when there is only one signer in the recording.

Although MediaPipe poses are available from WMT-SLT22, we re-extract them with *pose-format* (Moryossef, 2022). This library defines a standardized way of storing poses, and provides different functionalities to work with them. After the extraction, we obtain an array with the same shape as described in Section 2.

While the video recordings in the SRF dataset have a rate of 25 frames per second (fps), the FocusNews dataset present a frame rates of either 25, 30 or 50 fps. We perform cubic interpolation for the extracted poses, to achieve a unified frame rate of 25fps, using the *interpolate_fps* function from *pose-format*.

We build an independent vocabulary with the training split of each dataset. For the SRF dataset, organizers provide parallel and monolingual data. The latter contains all German subtitles, including much more sentences than the former. Therefore, we choose to build our SRF vocabulary from the monolingual data. Note that this data does not have paired (or parallel) videos, so it can not be used for training the model.

3.2 Architecture

We build a smaller Transformer architecture than the WMT-SLT22 baseline, since we observed signs of overfitting when checking the training losses. With the baseline architecture the system was simply generating the most frequent words from the training set. Therefore, we concluded that training a smaller model would hinder the overfitting and might improve the results.

In particular, our Transformer model has a symmetrical structure for the encoder and decoder, with 3 layers, 4 heads, 1024 neurons in the feed forward network layers, and an embedding dimension of 256.

3.3 Data augmentation

The performance of deep learning models depends on the quality, quantity, and domain of training data, however datasets that provide all qualities needed for models are often not available. To diminish the consequences of the data scarcity, a common practice is to apply data augmentation techniques.

[2] https://github.com/mt-upc/fairseq/tree/wmt-slt22

This approach is really useful to improve the performance of models, and makes them more robust to slight changes in the input data.

We made use of the *augment2d* function from the *pose-format* library, which allows applying various transformations directly to the keypoints, such as random rotation, shear effect, and scaling. Specifically, the rotation angle in radians, the shear factor and the scaling factor we apply are obtained by sampling from a normal distribution with zero mean and standard deviation of 0.2. Some examples of augmented poses are shown in Figure 1.

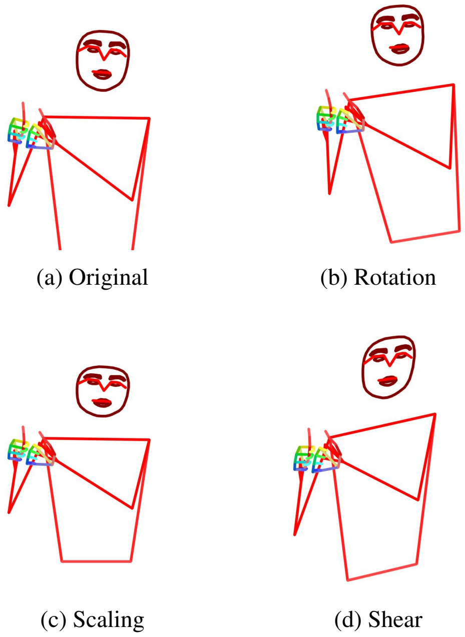

(a) Original
(b) Rotation
(c) Scaling
(d) Shear

Figure 1: Data augmentation transformations.

3.4 Vocabulary size

State-of-the-art Machine Translation systems use subword dictionaries instead of word-level vocabularies (Tran et al., 2021; Yang et al., 2021). These dictionaries are built by decomposing words into smaller pieces based on their frequency (Sennrich et al., 2016). Analogously to the baseline approach, we use SentencePiece tokenizer to obtain the subword vocabulary (Kudo and Richardson, 2018).

The vocabulary size is a hyperparameter that, in practice, is either chosen arbitrarily or via trial-and-error (Salesky et al., 2020). However, it has been studied that using a greater vocabulary size might help in reducing the class imbalance present in the training dataset (Gowda and May, 2020). The baseline used a vocabulary size of 1000 subwords, and we experiment with 2000 and 4000. Our goal was to detect whether downsampling a vocabulary of 20k unique words, for FocusNews dataset, to 1k subwords may be oversimplifying the problem.

3.5 Pretraining with PHOENIX14-T

We also explore transfer learning to overcome the data scarcity problem. Given that the scope of WMT22 is on Swiss German Sign Language, we chose the PHOENIX14-T dataset (Camgoz et al., 2018). For the three datasets, the target language is either spoken (written) German or Standard German from Switzerland. However, PHOENIX14-T presents an important domain shift with respect to the WMT-SLT22 datasets, since it is limited to live interpretation of weather forecast on broadcast TV.

We pretrained our model with the PHOENIX14-T dataset. In order to implement the transfer learning pipeline, we built a vocabulary by merging the training data from the three available datasets: FocusNews, SRF and PHOENIX14-T.

3.6 Checkpoint Averaging

We choose the best-performing models as those with the best BLEU dev scores. However, in the best-performing cases, and as thee final step in our trainings, we average the weights of the 3 best model checkpoints for each run. This methodology, which was firstly introduced by (Vaswani et al., 2017), proved to be a useful and easy to implement technique to generate more robust predictions on Transformers (Popel and Bojar, 2018), and has been widely used in the Machine Translation field.

4 Results

We train our systems with FocusNews and SRF. For both datasets, we provide results for the dev set, which contains recordings from FocusNews and SRF. We choose the 6 best-performing models for dev to submit to the official challenge submission.

Table 2 shows the results of models trained with FocusNews. We notice an improvement in the performance of our baseline implemented iin Fairseq with respect to the one from the organizers (1-2). We then analyze the effect of the vocabulary size based on the BLEU obtained for the dev set. We notice that using 4000 subwords provides the best results in terms of vocabulary size (2-4). Therefore, we choose this configuration to experiment with pretraining and data augmentation. For this set of experiments, we see a slight improvement when fine-tuning a network that has been pretrained with the PHOENIX14-T dataset (6). However, adding data augmentation (5) or a combination of both

ID	System	Vocab. size	Data Augm.	Pretrain	BLEU (dev)	BLEU (test)
1	Baseline (Müller et al., 2022)	1k			0.22	-
2	Our Baseline (§3.2)	1k			0.47	**0.50**
3	2 + 2k subwords	2k			0.47	-
4	2 + 4k subwords	4k			0.62	-
5	4 + data augmentation	4k	✓		0.51	-
6	4 + pretrain w/ Phoenix	4k		✓	**0.64**	0.41
7	6 + data augmentation	4k	✓	✓	0.48	-
8	6 + checkpoint average	4k		✓	0.57	0.35

Table 2: Results of models trained with the FocusNews dataset. *BLEU (dev)* corresponds to the results obtained in the challenge dev set, and *BLEU (test)* to the results extracted by the organizers using the official test partition (Müller et al., 2022). In bold are the best results for each partition.

ID	System	Vocab. size	Data Augm.	Pretrain	BLEU (dev)	BLEU (test)
1	Baseline (Müller et al., 2022)	1k			0.59	0.12
2	Baseline (§3.2)	1k			0.64	-
3	2 + 2k subwords	2k			**0.69**	**0.28**
4	2 + 4k subwords	4k			0.63	**0.28**
5	3 + checkpoint average	2k			0.60	0.24

Table 3: Results of models trained with the SRF dataset. *BLEU (dev)* corresponds to the results obtained in the challenge dev set, and *BLEU (test)* to the results extracted by the organizers using the official test partition (Müller et al., 2022). In bold are the best results for each partition.

data augmentation and pretraining (7) does not improve the results. Checkpoint averaging does not bring an improvement either (8). Surprisingly, we cannot extract the same conclusions from the test results. After we received the preliminary findings from the organizers (Müller et al., 2022), we found that the best-performing model for dev (6) was not the best for test, but the simplest one (2).

Results of models trained with SRF are presented in Table 3. Similarly to FocusNews, our proposed baseline improves the organizers' in this dataset (1-2). However, in this case, the optimal number of subwords is 2000, with a slight improvement over other vocabulary sizes (2-4). Similar to the FocusNews case, we observe that checkpoint averaging does not improve results. Due to technical issues and time limitations, experiments with SRF are limited to analyzing the vocabulary size, hence the best experiment with FocusNews could not be replicated using SRF. Although results for dev are better for models trained with SRF, results for test show a poorer performance than FocusNews models.

We optimized our systems to obtain the best BLEU metric, without taking other metrics into consideration. However, organizers also compute chrF++ (Popović, 2017) and BLEURT (Sellam et al., 2020) metrics (Müller et al., 2022). We find that the BLEURT score shows a similar performance than BLEU. However, for the chrF++ metric, which correlates better with respect to human relative rankings, our models score lower compared to other submissions.

We provide some examples of the sentences generated by our best-performing model in Table 4. We see that the translations are poor and lack correlation with the video, which relates to the poor overall performance in the BLEU metric.

5 Discussion and Conclusion

We proposed a pipeline to tackle the Sign Language Translation Task for the newly released datasets:

Ref.:	*Letztes Jahr haben viele Gehörlosen-Medien über die erste Gehörlosen Universität in Europa in Bad Kreuznach in Deutschland berichtet.*
Pred.:	*Am letzten Samstag, 22. Mai, in der Schweiz, in der Schweiz, in der Schweiz, GSC Aarau, GSC Aarau.*
Ref.:	*Dazu sind 4 Politiker eingeladen, die über für Behinderte wichtige Themen diskutieren werden, wie zum Beispiel TV-Untertitel, UNO Konvention für Behinderte usw.*
Pred.:	*Das Ziel ist es, dass es, dass die nempflichkeit für die nempflichkeitssetzen kann.*
Ref.:	*Der Deutsche Fernsehsender ZDF bietet Filme im Internet mit Untertitel an, sofern der Film vorher im Fernseher mit Untertitel ausgestrahlt wurde.*
Pred.:	*Der Schweizerische Gehörlosenbund SGB-FSS organisiert mit dem Schweizerischen Gehörlosenbund SGB-FSS, der Gehörlosen Sportverband der Gehörlosen Sportverband der Gehörl osen Sportverband der Gehörlosen Sportverband der Gehörlosen Sportverband der Gehörlosen Sportverband.*

Table 4: Example reference and predictions from our best-performing model for the official dev partition.

Focusnews and SRF. Our fresh implementation with Fairseq slightly improved the baseline provided by the organizers.

Our findings showed that when training with FocusNews, our baseline system has the best performance for test. The changes in vocabulary size did not affect the test performance. Furthermore, we showed that using checkpoint averaging does not help for this task. In all cases, we still think that the results are extremely low, which indicate a really poor translation, and there is potential unreliability of the findings due to the close to 0 BLEU score.

We consider the results we obtained can be further improved, so we leave some experiments for future work. Firstly, we believe that a joint training from the two provided datasets could boost the performance of the models by bridging the domain gap between these datasets. Secondly, we did not see any improvement by pretraining the models with PHOENIX14-T dataset. However, we think that solving the WMT-SLT22 task must require some sort of transfer learning from a pretrained model.

Limitations

As stated by the organizers, results are still poor. When inspecting the predictions, it seems evident that the model is learning the most frequent words in the vocabulary, thus failing to provide meaningful predictions from the video. We consider that this is due to the high complexity of the task paired with a lack of data.

	Length	**Words**	**Ratio**
FocusNews	19 h	21 k	0.90
SRF	16 h	19 k	0.84
How2Sign	79 h	16 k	4.93
PHOENIX14-T	11 h	3 k	3.67

Table 5: Comparison between SLT datasets based on the duration of the videos (in hours) and number of unique words (in thousands) in the vocabulary. The Ratio column provides an indication of the difficulty of solving the SLT task for each dataset.

As shown in Table 5, the ratio between the training data and vocabulary size is much lower compared to other SLT datasets such as How2Sign and PHOENIX14-T. We take these results as an indication of the complexity of the datasets. We hypothesize that the low BLEU scores reported in the baseline, may be caused by the low ratio between video hours per unique words in the vocabulary, hence the dataset might be too complex. Therefore, we decide to experiment with data augmentation since it artificially improves the amount of training samples.

Our experiments with the SRF dataset, have been computationally expensive. Due to technical details, we have to read full sequences of around 30 minutes every time we load a sample. Processing them, even with the dimensional reduction provided by pose estimators, has been a challenge for the machines of our academic lab. A slightly better set of results might have been produced, but we still think it would not have made a significant difference.

In addition, we present the input poses as a sequence of one-dimensional arrays with the XYZ coordinates. We think that this may not be the optimal way of processing the graph-like structure from poses. Using graph neural networks to preprocess input poses (Yan et al., 2018, 2019; Bull et al., 2020; Jiang et al., 2021) might be an interesting approach to improve SLT results.

We also lacked the time to experiment with other features, such as processing the RGB videos with a convolutional network (Vaezi Joze and Koller, 2019; Li et al., 2020; Albanie et al., 2020). We tried extracting i3d features fine-tuned on PHOENIX14-T, but the output features contained an excessive number of 0's, and we never run a proper experiment with this setup. This might happen because the visual appearance of the videos is too different between PHOENIX14-T and the WMT-SLT22 datasets, specifically due to the spatial segmentation of the signer in the frames provided in these datasets.

Acknowledgements

This research was partially supported by research grant Adavoice PID2019-107579RB-I00 / AEI / 10.13039/501100011033, research grants PRE2020-094223, PID2021-126248OB-I00 and PID2019-107255GB-C21.

References

Samuel Albanie, Gül Varol, Liliane Momeni, Triantafyllos Afouras, Joon Son Chung, Neil Fox, and Andrew Zisserman. 2020. BSL-1K: Scaling up co-articulated sign language recognition using mouthing cues. In *European Conference on Computer Vision (ECCV)*. Springer.

Hannah Bull, Michèle Gouiffès, and Annelies Braffort. 2020. Automatic segmentation of sign language into subtitle-units. In *European Conference on Computer Vision*, pages 186–198. Springer.

Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural sign language translation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7784–7793.

Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. 2019. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Amanda Duarte, Samuel Albanie, Xavier Giró-i Nieto, and Gül Varol. 2022. Sign language video retrieval with free-form textual queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14094–14104.

Amanda Duarte, Shruti Palaskar, Lucas Ventura, Deepti Ghadiyaram, Kenneth DeHaan, Florian Metze, Jordi Torres, and Xavier Giro-i Nieto. 2021. How2Sign: A Large-scale Multimodal Dataset for Continuous American Sign Language. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Thamme Gowda and Jonathan May. 2020. Finding the optimal vocabulary size for neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3955–3964, Online. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation.

Songyao Jiang, Bin Sun, Lichen Wang, Yue Bai, Kunpeng Li, and Yun Fu. 2021. Skeleton aware multimodal sign language recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. 2020. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1459–1469.

Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. 2019. Mediapipe: A framework for building perception pipelines. *ArXiv*, abs/1906.08172.

Amit Moryossef. 2022. Complete toolkit for working with poses. `https://github.com/AmitMY/pose-format/`.

Mathias Müller, Sarah Ebling, Necati Cihan Camgöz, Zifan Jian, Alessia Battisti, Katja Tissi, Sandra Sidler-Miserez, Regula Perrollaz, Michèle Berger, Sabine Reinhard, Amit Bar-Ilan University Moryossef, Annette Rios, Richard Bowden, Ryan Wong, Robin Ribback, and Severine Schori. 2022a. WMT-SLT SRF: Training data for the WMT shared task on sign language translation (videos, subtitles). We additionally acknowledge funding through the Innosuisse Flagship "Inclusive Information and Communication Technologies" (IICT) (grant agreement no. PFFS-21-47).

Mathias Müller, Sarah Ebling, Necati Cihan Camgöz, Zifan Jiang, Alessia Battisti, Amit Moryossef, Annette Rios, Richard Bowden, and Ryan Wong. 2022b. WMT-SLT FocusNews: Training data for the WMT shared task on sign language translation. We additionally acknowledge funding through the Innosuisse Flagship "Inclusive Information and Communication Technologies" (IICT) (grant agreement no. PFFS-21-47).

Mathias Müller, Sarah Ebling, Avramidis Eleftherios, Alessia Battisti, Michèle Berger, Richard Bowden, Annelies Braffort, Necati Cihan Camgöz, Cristina España-Bonet, Roman Grundkiewicz, Zifan Jiang, Oscar Koller, Amit Moryossef, Regula Perrollaz, Sabine Reinhard, Annette Rios, Dimitar Shterionov, Sandra Sidler-Miserez, Katja Tissi, and Davy Van Landuyt. 2022. Findings of the WMT 2022 shared task on sign language translation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mathias Müller, Annette Rios, and Amit Moryossef. 2022. Sockeye baseline models for sign language translation. `https://github.com/bricksdont/sign-sockeye-baselines`.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Martin Popel and Ondřej Bojar. 2018. Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110:43–70.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Elizabeth Salesky, Andrew Runge, Alex Coda, Jan Niehues, and Graham Neubig. 2020. Optimizing segmentation granularity for neural machine translation. *Machine Translation*, 34(1):41–59.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. Facebook AI's WMT21 news translation task submission. In *Proceedings of the Sixth Conference on Machine Translation*, pages 205–215, Online. Association for Computational Linguistics.

Hamid Vaezi Joze and Oscar Koller. 2019. Ms-asl: A large-scale data set and benchmark for understanding american sign language. *The British Machine Vision Conference (BMVC)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Andreas Voskou, Konstantinos P. Panousis, Dimitrios Kosmopoulos, Dimitris N. Metaxas, and Sotirios Chatzis. 2021. Stochastic transformer networks with linear competing units: Application to end-to-end sl translation.

Sijie Yan, Yuanjun Xiong, Wangm Jingbo, and Dahua Lin. 2019. Mmskeleton. `https://github.com/open-mmlab/mmskeleton`.

Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*.

Jian Yang, Shuming Ma, Haoyang Huang, Dongdong Zhang, Li Dong, Shaohan Huang, Alexandre Muzio, Saksham Singhal, Hany Hassan, Xia Song, and Furu Wei. 2021. Multilingual machine translation systems from Microsoft for WMT21 shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 446–455, Online. Association for Computational Linguistics.

Separating Grains from the Chaff: Using Data Filtering to Improve Multilingual Translation for Low-Resourced African Languages

Idris Abdulmumin[1,2*], **Michael Beukman**[3*], **Jesujoba O. Alabi**[4*], **Chris Emezue**[5,6],
Everlyn Asiko[7,8], **Tosin Adewumi**[9], **Shamsuddeen Hassan Muhammad**[2,10],
Mofetoluwa Adeyemi, Oreen Yousuf[11], **Sahib Singh**[12], **Tajuddeen Rabiu Gwadabe**[2,13]

All: MasakhaneNLP, [1]Ahmadu Bello University, Zaria, Nigeria, [2]HausaNLP, [3]University of the Witwatersrand, South Africa,
[4]Saarland University, Germany, [5]TUM, Germany, [6]Mila - Quebec AI Institute, [7]University of Cape Town, South Africa,
[8]African Institute for Mathematical Sciences, [9]Luleå University of Technology, Sweden, [10]LIAAD-INESC TEC, Porto, Portugal,
[11]Uppsala University, Sweden, [12]Ford Motor Company, [13]University of Chinese Academy of Sciences, China

iabdulmumin@abu.edu.ng

Abstract

We participated in the WMT 2022 Large-Scale Machine Translation Evaluation for the African Languages Shared Task. This work describes our approach, which is based on filtering the given noisy data using a sentence-pair classifier that was built by fine-tuning a pre-trained language model. To train the classifier, we obtain positive samples (i.e. high-quality parallel sentences) from a gold-standard curated dataset and extract negative samples (i.e. low-quality parallel sentences) from automatically aligned parallel data by choosing sentences with low alignment scores. Our final machine translation model was then trained on filtered data, instead of the entire noisy dataset. We empirically validate our approach by evaluating on two common datasets and show that data filtering generally improves overall translation quality, in some cases even significantly.

1 Introduction

This paper presents Masakhane NLP's submission to the WMT 2022 large-scale machine translation evaluation for African languages. We participated in the constrained translation task and chose to focus on a subset of all the language pairs considered for this task due to resource constraints. We specifically explore the language directions **{hau, ibo, lug, swa, tsn, yor, zul}**↔**eng** and **wol**↔**fra**, and submitted our primary and secondary systems which were competitive with other submissions for this task.

Machine translation has received much attention recently, especially for low-resourced languages (Adelani et al., 2022a; Fan et al., 2021; Haddow et al., 2022; Hoang et al., 2018; Nekoto et al., 2020). A promising approach for such setups is to fine-tune large pre-trained language models on the available small amount of translation data (Neubig and Hu, 2018; Adelani et al., 2021a, 2022a). While most of these language models are trained on predominantly high-resourced language datasets (Conneau et al., 2020; Devlin et al., 2019; Radford et al., 2018), there have been a few models that were pre-trained (Ogueji et al., 2021) or adaptively fine-tuned (Alabi et al., 2022) only on low-resourced languages.

Recent works have tried, successfully, to supplement the existing small amounts of natural data in low-resource languages with artificially generated parallel data. For instance, in machine translation, Sennrich et al. (2016) and Ueffing (2006) padded the true parallel data with automatic translations of monolingual sentences through back-translation and self-learning respectively. Others, such as Bañón et al. (2020); El-Kishky et al. (2020); and Schwenk et al. (2021), have used different approaches for detecting potentially aligned sentences within web datasets. While significant improvements have been achieved with these synthetic datasets, an in-depth investigation by Kreutzer et al. (2022) has found them to be fraught with many issues, such as misalignment, wrongful language codes, etc.

Similarly, research has shown that data quality plays an important role in the performance of natural language processing (NLP) models, in machine translation specifically (Arora et al., 2021; Dutta et al., 2020; Hasan et al., 2020; Tchistiakova et al., 2021), but also more generally in other NLP tasks (Abdul-Rauf et al., 2012; Alabi et al., 2020). It was found that often times, models that were trained on smaller amounts of high-quality data outperform their counterparts that are trained on larger amounts of noisy datasets (Gascó et al., 2012; Przystupa and Abdul-Mageed, 2019; Abdulmumin et al., 2022; de Gibert et al., 2022). This has led to many studies (Eetemadi et al., 2015) and prior WMT tasks (Koehn et al., 2018, 2019, 2020) that

[*] Equal contribution.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1001 - 1014
December 7-8, 2022 ©2022 Association for Computational Linguistics

attempt to find ways to improve the quality of existing data, which, as mentioned before, is often rife with errors.

Therefore, in our submission to the shared task, we experimented with filtering web-mined data for African languages using pre-trained language models and evaluated the effect of using this filtered data on machine translation performance. We defined our filtering approach as a sentence-pair binary classification task and fine-tuned a pre-trained language model using positive and negative samples. We used sentences from the high-quality MAFAND-MT (Adelani et al., 2022a) dataset (which was included in the training data for the constrained task) as positive examples and created negative examples by extracting sentences with low language-agnostic sentence representations (LASER) (Artetxe and Schwenk, 2019b) alignment scores from the `wmt22-african` (NLLB Team et al., 2022) corpus that was provided for this task. Our results highlight the importance of filtering on the quality of the final machine translation system. We also detail how to create a high-quality filter for African languages using a few gold-standard parallel sentences. We release our codes on GitHub.[1]

The rest of the paper is organized as follows: in Section 2, we review related work, and in Section 3, we present the dataset we used. Section 4 provides an overview of the bitext filtering approach, while Section 5 details experimental settings and the translation model architecture. In Section 6, we evaluate the model's performance, and lastly in Section 7, we conclude and highlight some future research directions.

2 Related Work

One of the difficulties when dealing with low-resourced settings, as we do here, is that high-quality parallel texts are particularly scarce (Koehn and Knowles, 2017). To curate data for such language pairs, methods for automatically mining parallel text from the web using heuristics (Resnik, 1999) or latent space and similarity-based filters (Artetxe and Schwenk, 2019a; Schwenk et al., 2021) have been proposed. These have led to the curation of publicly available web-mined datasets such as CCAligned (El-Kishky et al., 2020), CC-Matrix (Fan et al., 2021; Schwenk et al., 2021), ParaCrawl (Esplà et al., 2019), and WikiMatrix (Schwenk et al., 2019) to mention just a few.

However, the recent research work by Kreutzer et al. (2022) shows that the automatically aligned and mined parallel bitexts, especially for low-resource language pairs, contain various degrees of errors and less than half of the data are of good quality. Additionally, many approaches generate large amounts of synthetic data, often through back-translation, where synthetic parallel data is generated by automatically translating monolingual data (Bojar and Tamchyna, 2011; Lambert et al., 2011; Sennrich et al., 2016). While additional data has the potential to improve the trained models, these synthetic datasets are often of low quality (Xu et al., 2019). These observations have led to an increased interest in the automatic filtering of noisy bitexts as a key research topic in machine translation (MT).

One approach to improve data quality is to filter out the noisy or invalid parts of a large corpus, keeping only a high-quality subset thereof (Abdulmumin et al., 2021). In this vein, numerous filtering methods have been developed (Axelrod et al., 2011; Eetemadi and Toutanova, 2015; Junczys-Dowmunt, 2018). For instance, Xu et al. (2019) use the cosine similarity between sentence embeddings as a measure of how closely aligned two sentences are. Imankulova et al. (2017) perform back-translation and then filter based on the sentence-level BLEU score, keeping only those sentences with a high BLEU. Similarly, Adjeisah et al. (2021) perform a round-trip translation and only use the sentence pair if it is sufficiently close to the original sentence, according to a chosen similarity measure. There has also been work on alignment between two parallel corpora, and Hasan et al. (2020) uses the LASER score[2] to evaluate alignment, and filter out all sentences below a specific threshold.

3 Datasets

We participated in the constrained translation track and used only the provided dataset. We present the various dataset used, their sizes and corresponding sources in Table 9 in Appendix A. For our experiment, we selected 8 language pairs and developed different multilingual machine translation systems for them. These language pairs are **{hau, ibo, lug, swa, tsn, yor, zul}**↔**eng** and **wol**↔**fra**. According to the recommendation

[1] https://github.com/abumafrim/WMT22-MasaKhane

[2] https://github.com/facebookresearch/LASER

Direction	Parallel sentences	Problem
eng → hau	**src:** I booked the house for my husband's family as we were getting married in Ericeira. **tgt:** na tsarr da aba a ka kasarr ni ila ure imbarr yi ngbangbamu.	**tgt** is not a Hausa sentence
eng → hau	**src:** "Go hunt, and may the light be with you.""" **tgt:** """Zo, zo muje, ke kika hada fitinar ke za ki warware ta."""	**tgt** is not a translation of the **src**
eng → hau	**src:** The Moslem creed. **tgt:** Musa Aminta	mismatched named entities
eng → hau	**src:** Israel **tgt:** oooooooooooooooooooooooooooooooooooooooooooooooo ooooooooBBBBBBBBBBBBBBBBBBBBBBBB	mistranslation; foreign characters

Table 1: Examples of noise in the auto-aligned bitext

of Kreutzer et al. (2022), we carefully examined the training dataset provided by manual inspection and divided it into two categories based on the source of the data and the amount of noise included therein. In the following subsections, we describe these two categories of data.

3.1 Clean Bitext

This category of training data comprises all the datasets that are considered to be manually curated. The datasets in this category include: bibleuedin (Christodouloupoulos and Steedman, 2015), MAFAND-MT,[3] QED (Abdelali et al., 2014), Mozilla-I10n,[4] Tanzil,[5] and several others listed in Table 9. The clean bitext consists of sentences mostly in the news and religious domains, with a few in the health, education, and technology domains. We also refer to the clean bitext as True Parallel in this paper.

3.2 Noisy Bitext

We categorized all the automatically aligned bitext as noisy bitext. This also includes the LASER filtered data. The sentences in this category make up the majority of the training dataset, making up 99.2% of the total training data. The datasets in this category include: CCAligned, CCMatrix, LASER wmt22_african,[6] WebCrawl African,[7] and the following datasets from OPUS (Tiedemann, 2012): MultiCCAligned (El-Kishky et al., 2020), TED2020 (Reimers and Gurevych, 2020), WikiMatrix (Schwenk et al., 2019), XLEnt (El-Kishky et al., 2021) and others highlighted in Table 9.

Language pair		Data size	% of original
eng	hau	9, 122, 559	99.9
	ibo	520,544	99.6
	lug	3,511,275	99.8
	swa	32,898,533	99.6
	tsn	6,036,656	99.1
	yor	1,718,105	99.3
	zul	4,142,146	97.6
fra	wol	237,348	100.0

Table 2: Training data after filtering using heuristics

On manual inspection, however, we found numerous issues with the data, including non-parallel sentences, sentences that consist of only numbers and/or punctuation, sentences in different languages, etc. Examples of noise in the auto-aligned data can be seen in Table 1.

3.3 Validation and Test Data

For the optimization of our translation systems, we combined the FLORES-101 (Goyal et al., 2022) and MAFAND-MT (Adelani et al., 2022a) development sets for each of the 8 language pairs. To compare the performance of the developed MT engines, we evaluated on the FLORES-101 devtest set and the MAFAND-MT test set.

4 Parallel Data Filtering

To attempt to deal with the highly noisy data, we opted to use filtering techniques to remove many invalid or incorrectly aligned sentences, similar to prior work (Arora et al., 2021; Hasan et al., 2020; Xu et al., 2019). We first used some simple heuristic approaches, described in Section 4.1, and then progress to an automatic filtering method, detailed in Section 4.2.

4.1 Heuristics

We filtered sentences that consist of only numbers and/or punctuation marks. After filtering, the statis-

[3] https://github.com/masakhane-io/lafand-mt.git

[4] https://github.com/mozilla-l10n/mt-training-data

[5] https://tanzil.net/trans/

[6] https://huggingface.co/datasets/allenai/wmt22_african

[7] https://github.com/pavanpankaj/Web-Crawl-African

Data	eng							fra
	hau†	ibo†	lug†	swa†	tsn†	yor†	zul†	wol†
Train	6,198	13,998	8,152	61,566	4,202	13,290	7,002	6,722
Dev	2,602	3,002	3,002	3,584	2,686	3,090	2,480	3,014
Test	3,002	3,002	3,002	3,672	3,002	3,118	1,998	3,002

Table 3: Sentence-pair classification training data: a mixture of MAFAND-MT† sentence pairs, taken as positive samples, and wmt22_african (worst pairs based on LASER scores), taken as negative samples.

tics of the resulting training dataset are shown in Table 2. The table shows that 2.4% of the original Zulu (zu) data consisted of just numbers or punctuation, while other languages had smaller invalid portions, between 0.0% and 0.1%.

4.2 Automatic Filtering

Due to the large size of the automatically aligned dataset, we adopted an automatic approach to determine the quality of parallel sentences to train our translation models. The approach we adopted is sentence-pair binary classification (Nguyen et al., 2021), where we used a transformer-based model to predict the probability that two aligned sentences are actual translations of each other. We explain the process of training data generation and the experimental choices for building the filtering model.

4.2.1 Positive and negative samples

To create the training and evaluation data for the sentence-pair classification-based filtering, we generated positive and negative samples from the training data available for this task. We used the train, dev and test sets from the MAFAND-MT dataset, which is a gold-standard parallel dataset, as positive examples. For the negative examples, however, we sorted the sentences in wmt22_african dataset that was provided for this task based on their LASER alignment scores, and selected the least scored sentences in equal amounts to each of the positive examples. The distribution of the train, dev and test samples is presented in Table 3.

4.2.2 Model

We fine-tuned two pre-trained language models, ALBERT (Lan et al., 2020) and AfroXLMR (Alabi et al., 2022) for the sentence pair binary classification task. ALBERT was selected based on its performance on downstream NLP tasks (Lan et al., 2020), even though it has fewer parameters than other BERT-based models (Nguyen et al., 2021). AfroXLMR, on the other hand, was chosen because it was trained on African languages (Alabi et al., 2022), and such a setup has been shown to improve performance on downstream tasks for these languages (Adelani et al., 2022a).

4.3 Filter Training Setup

The filtering models were trained to accept a pair of sentences from the source and target languages. During training, the [CLS] token hidden representation of the input sentence pairs is fed into a linear Layer and the model is optimized using binary cross entropy loss. However, at inference time, we add a sigmoid layer to the output to predict a number between 0.0 and 1.0 indicating the likelihood of the bitexts being translations of each other. We fine-tuned these models using each language's train split of positive and negative samples, then evaluated performance on the test set while optimizing on the development set.

The performance of the various automatic filtering models and the subsequent sizes of the filtered datasets for the 8 language pairs are shown in Table 4. This table shows the number of sentence pairs the models classified as actual translation pairs using a threshold of 0.5 and 0.7 as well as the F1 score when using the 0.5 threshold. Additionally, in Table 5, we show the number of sentences that were classified by two or all three of the models as being high-quality.

5 MT Experiments

To evaluate the effect of our filtering techniques, we trained some multilingual NMT models for the 8 language pairs that we have selected for this task. In the following subsections, we highlight the model architectures, training setups, and different multilingual models that were trained.

5.1 Model Architecture

For our experiments, we fine-tune M2M-100 (Fan et al., 2021) on different subsets of the provided data. M2M-100 is a pretrained translation model trained on several languages including African languages, as such it has seen all the languages we have chosen for this task during pre-training. We use the model with $418M$ parameters.

5.2 Training Setup

We fine-tuned the M2M-100 model based on the implementation within the Fairseq[8] toolkit (Ott

[8] https://github.com/facebookresearch/fairseq

Model		en							fr	$F1_{avg.}$
		hau	ibo	lug	swa	tsn	yor	zul	wol	
ALBERT-base	F1	95.6	94.2	94.7	89.6	95.7	91.1	87.4	95.1	92.9
	t=0.5	278,930	78,056	119,516	5,832,820	346,329	151,886	363,739	6,552	
	t=0.7	197,232	63,207	82,243	3,921,959	252,499	91,366	213,991	4,365	
ALBERT-xlarge	F1	93.2	92.8	96.3	63.7	95.3	90.7	89.1	84.4	88.2
	t=0.5	115,987	129,304	146,948	3,263,429	273,154	113,860	613,483	49,926	
	t=0.7	81,641	111,562	102,354	1,638,528	217,200	86,558	302,951	41,283	
AfroXLMR-base	F1	96.9	94.4	95.4	94.6	96.1	98.4	88.0	97.1	95.1
	t=0.5	296,881	75,102	149,051	6,139,327	363,155	81,902	281,803	6,997	
	t=0.7	226,666	59,995	84,499	5,064,365	276,490	73,786	171,778	5,189	

Table 4: Training data after filtering using sentence-pair classifier — t=Threshold; F1 was computed at t=0.5

t=0.5	Albert-base	Albert-xlarge	AfroXLMR
Albert-base	2,984,862	1,750,707	2,575,408
Albert-xlarge	-	2,107,204	1,058,711
AfroXLMR	-	-	3,925,612
sents. in ALL			**668,633**
t=0.7			
Albert-base	1,977,486	909,203	1,884,922
Albert-xlarge	-	1,206,493	547,925
AfroXLMR	-	-	3,420,147
sents. in ALL			**331,208**

Table 5: Data overlap after filtering using the sentence-pair classifier models

et al., 2019). We used batch sizes of $2,048$ tokens, a maximum sentence length of $1,024$, and a dropout of 0.3. For optimization, we used Adam (Kingma and Ba, 2015) with β_1 = 0.9 and β_2 = 0.998, a learning rate of $5e - 5$ and a warmup of $2,500$ updates. The optimizer uses a label-smoothed cross-entropy loss function with a label-smoothing value of 0.2. All models were trained for a maximum of $1,000,000$ update steps. We tokenized all data using the model's SentencePiece (Kudo and Richardson, 2018) tokenizer.

To evaluate our models and to choose the best checkpoints, we used the BLEU score (Papineni et al., 2002) calculated with the SacreBLEU (Post, 2018) implementation. In addition, we also evaluated the models using CHRF (Popović, 2015).

5.2.1 Baseline models

We train many-to-many (M2M) translation models by fine-tuning M2M-100 on the following subsets of the datasets described in Section 3. These include, the clean bitexts described in Section 3.1, noisy bitext described in Section 3.2, and a mixture of the clean and noisy bitexts. The noisy bitext was only partially cleaned, as evidenced in Table 2, using the heuristic rules mentioned in Section 4.1 without applying the proposed automatic filtering on data.

We trained these baseline models to compare and measure the efficacy of our filtering technique on the quality of the translation models. We submitted the model in (i) as our secondary system for this task.

5.2.2 Models on filtered data only

To evaluate the effect of the filtered data on the quality of the translation output, we train M2M models on the filtered data from the different models using a threshold of 0.5 and 0.7.

5.2.3 Models on filtered and clean data

We went further to train multilingual models on the concatenation of the noisy and clean text, and on the filtered and clean data for easier comparison. With this system, we were able to measure the amount of improvement we can obtain by including the clean bitext compared to training models only on the filtered bitext.

6 Results and Discussion

In Tables 6 and 7, we report the BLEU and CHRF scores obtained by the different models that we trained, as evaluated on the FLORES-101 devtest and MAFAND-MT test datasets, respectively.

6.1 Baseline Models

On average, the baseline model trained on the clean bitext performed impressively on the two evaluation datasets, despite the limited dataset size. On MAFAND-MT, the model trained on the clean bitext obtained a higher BLEU score than the model trained on the noisy bitext, and on FLORES-101, the reverse was true. This is likely due to the fact that the MAFAND-MT data is present in the clean

Models	eng→x							fra→x	x→*eng*							x→*fra*	Avg.
	hau	ibo	lug	swa	tsn	yor	zul	wol	hau	ibo	lug	swa	tsn	yor	zul	wol	
BLEU																	
Baselines																	
Clean bitext	9.30	13.19	**4.00**	23.17	8.56	**3.60**	9.43	**3.56**	14.24	12.56	11.24	26.86	8.78	8.90	18.51	6.03	11.37
Noisy bitext	15.32	10.77	2.14	30.64	12.87	2.57	12.35	0.69	20.58	14.69	13.19	31.80	16.29	11.40	24.68	3.22	13.95
Clean + Noisy bitext	15.34	11.37	2.40	30.48	13.31	2.48	12.61	0.73	20.53	15.07	13.34	31.61	16.50	11.75	24.29	3.88	14.11
Filtered only																	
albert-xlarge-0-7	16.43	15.38	2.54	29.89	**16.31**	3.00	15.18	0.65	20.05	17.32	12.51	34.24	18.55	12.62	27.31	5.14	15.45
Filtered + Clean bitext																	
albert-xlarge-0-5	16.05	15.01	3.22	**33.31**	15.96	3.08	14.97	1.99	**20.92**	17.45	13.93	**34.99**	18.24	13.24	27.65	6.43	16.03
albert-xlarge-0-7	16.55	**15.70**	3.45	31.97	**16.31**	3.16	**15.50**	2.12	20.85	**17.88**	**13.97**	34.40	**18.29**	**13.38**	**27.35**	**7.20**	**16.13**
CHRF																	
Baselines																	
Clean bitext	34.01	**45.31**	**42.14**	55.14	45.58	**30.56**	43.62	**30.55**	34.70	**45.20**	**46.21**	54.53	45.37	39.04	46.50	**30.77**	41.83
Noisy bitext	30.04	34.18	33.04	54.34	43.51	16.23	46.30	8.92	35.15	37.46	35.96	54.38	45.39	33.84	49.85	15.72	35.89
Clean + Noisy bitext	30.53	35.75	33.69	54.66	44.23	15.90	46.37	10.91	35.70	38.82	37.50	54.78	45.62	35.35	49.71	19.21	36.79
Filtered only																	
albert-xlarge-0-7	36.18	41.71	36.94	54.79	51.64	18.85	51.14	10.86	37.18	41.38	39.07	56.81	**56.81**	38.27	52.71	22.20	40.41
Filtered + Clean bitext																	
albert-xlarge-0-5	36.56	43.19	40.44	56.65	51.25	20.33	50.77	23.44	37.79	43.72	44.34	57.70	51.98	40.01	52.75	27.76	42.42
albert-xlarge-0-7	36.64	44.32	41.44	56.60	**52.98**	21.88	**51.43**	25.22	38.11	44.23	45.14	**57.73**	52.22	**40.68**	**53.02**	29.29	**43.18**

Table 6: Performance of the multilingual model on the **FLORES-101** devtest set, with the maximum BLEU per column in **bold**. x represents African languages.

bitext, and that the noisy bitext contains sentences that were taken from the web, including Wikipedia, which is the source of the FLORES-101 dataset. When we compared the model trained on the clean bitext to the model trained on the noisy bitext, we saw between a +1 and +2 improvement on FLORES-101 and between +5 and +8 improvement on MAFAND-MT for lug, wol, and yor. This confirms not only the importance of the data domain, but also the importance of data quality on the quality of the machine translation output.

After mixing the two datasets, the performance improved over using only the clean bitext by more than 6 BLEU on hau↔eng, and almost 3 BLEU on average across all languages on FLORES. The performance, though, was similar to using only the noisy bitext. On the MAFAND-MT test set, however, the performance deteriorated by almost 2 BLEU when compared to training on the clean bitext only. At language-pair level, eng→ibo was affected more (−9.14 BLEU), followed by eng→wol, whereas yor→eng benefited tremendously (+17.83 BLEU). On average, training on the two bitexts marginally improves over using only the noisy bitext, and this is consistent on all the test sets.

Investigating the results in more depth, we found that the BLEU scores of the models are lower when translating into an African language, similar to the findings of Adelani et al. (2022a). This effect is exacerbated for the languages with the fewest parallel sentences, such as lug, wol, and yor, except for ibo, which overall has the second-fewest parallel sentences, as shown in Table 9.

6.2 Data Filtering Analysis

We generally see that more filtering results in improved performance, corresponding to removing more noisy sentences from the data. Using less filtering, with a threshold of 0.5, generally performed slightly worse than using a threshold of 0.7. Both of these settings outperformed (a) using no filtering and (b) using no additional data.

We can also see the effect of the filtering steps on the training data in Tables 2 and 4. Filtering the data using heuristics resulted in only a small portion of the data being filtered out. Using the classifier, however, caused a large amount of noisy data to be removed. When looking at the F1 scores of the classification models, we can see that ALBERT-xlarge has the lowest F1, followed by ALBERT-base and AfroXLMR-base. Looking at Table 5, we can see that ALBERT-xlarge is also the most strict filter, removing the most data, whereas AfroXLMR-base removes the least amount of data. Interestingly, the number of sentences marked as high-quality by all three models is surprisingly low, possibly indicating that these different models (particularly ALBERT-xlarge and AfroXLMR-base) focus on different features of the data.

Finally, we saw that a higher threshold resulted in improved translation performance, but ALBERT-xlarge (which is quite strict) had a lower F1 than the

Models	eng→x							fra→x	x→*eng*							x→*fra*	Avg.
	hau	ibo	lug	swa	tsn	yor	zul	wol	hau	ibo	lug	swa	tsn	yor	zul	wol	
BLEU																	
Baselines																	
Clean bitext	**9.00**	**20.83**	**11.67**	25.81	18.64	**9.86**	14.50	**8.91**	**12.49**	**19.24**	20.00	29.28	20.44	16.98	23.20	**7.77**	16.79
Noisy bitext	5.24	10.37	6.12	25.35	16.61	3.61	15.23	0.98	8.52	12.83	14.35	28.37	21.34	13.14	26.74	1.57	13.15
Clean + Noisy bitext	5.59	11.69	6.54	25.55	17.25	3.42	15.10	1.99	8.80	13.64	15.67	28.67	21.74	**34.81**	26.68	2.33	14.97
Filtered only																	
albert-xlarge-0-7	7.75	16.33	7.56	26.45	23.01	4.59	17.63	0.86	9.93	15.59	16.77	30.92	30.92	16.46	29.47	3.09	16.08
Filtered + Clean bitext																	
albert-xlarge-0-5	8.49	18.16	10.11	**27.89**	22.99	5.37	17.68	5.46	11.73	17.53	20.63	32.38	27.07	17.84	29.88	5.52	17.42
albert-xlarge-0-7	8.74	19.08	10.26	27.80	**24.25**	6.09	**18.25**	6.05	12.32	17.58	**21.15**	**32.60**	**27.40**	18.54	**30.02**	6.77	**17.93**
CHRF																	
Baselines																	
Clean bitext	36.23	34.10	**31.59**	54.59	33.85	**21.97**	41.70	**26.22**	37.74	37.32	33.85	51.39	32.43	30.51	43.20	29.31	36.00
Noisy bitext	40.24	31.27	25.84	59.14	38.88	19.18	46.98	8.90	44.80	38.71	34.58	56.25	40.57	33.28	49.26	19.15	36.69
Clean + Noisy bitext	40.91	31.67	26.04	59.13	39.60	19.06	47.14	9.66	44.63	39.18	34.76	56.20	40.65	33.71	49.23	21.86	37.09
Filtered only																	
albert-xlarge-0-7	**44.19**	38.13	27.37	59.40	43.97	20.85	**51.96**	11.11	44.98	42.95	33.98	58.60	**43.12**	35.55	**52.09**	24.51	39.55
Filtered + Clean bitext																	
albert-xlarge-0-5	43.38	37.88	29.70	**61.47**	43.30	20.57	51.06	18.73	**45.53**	42.77	36.14	**58.93**	43.11	36.61	52.06	28.26	40.59
albert-xlarge-0-7	44.15	**38.72**	30.78	60.63	**44.11**	21.01	51.85	19.82	45.40	**43.31**	**36.15**	58.45	42.90	**36.81**	52.06	**29.52**	**40.98**

Table 7: Performance of the multilingual model on the **MAFAND-MT** test set, with the maximum BLEU per column in **bold**. x represents African languages.

other models, possibly suggesting that F1 performance does not fully indicate the expected downstream performance on the actual translation task.

6.2.1 The effect of filtering on translation models

We fine-tune M2M-100 for multilingual translation on the filtered data, and as expected, our results (on average) demonstrate a considerable improvement when the translation model is trained on the filtered data rather than the original noisy texts. In particular, for many languages, training on the filtered data from ALBERT-xlarge with a threshold of 0.7 outperformed the model trained on just the noisy bitext with at least a BLEU point.

Furthermore, we compared the performance of the model trained on only the clean data and on only the filtered data. Just as we saw with the baseline system, on MAFAND-MT, the model trained on the clean bitext performed better than the model trained on the filtered bitext, and on FLORES-101, the reverse was true. These results again confirm the importance of the filtering approach and further supports the observation that NMT engines are less robust to noise as found by Khayrallah and Koehn (2018), especially for low-resource settings.

6.3 Clean vs. filtered data

We find that on FLORES-101, adding in noisy, unfiltered data improves the results over just using the true parallel data. On MAFAND-MT, however, it generally reduces the BLEU score significantly. For both datasets, adding appropriately filtered data results in the highest performance averaged over all the languages, although for some specific languages, just using true parallel data resulted in the best performance.

Our performance on the test set provided by the organizers (Adelani et al., 2022b) is shown in Table 8. Here we can see that our primary model, which was trained on the clean bitext as well as the filtered data (filtered using ALBERT-xlarge, $t = 0.7$), significantly outperforms the model trained only on the clean bitext. We also see that our approach seems to have a larger performance gain when translating *from* African languages compared to translating *to* them.

7 Conclusion and Future Work

In this work, we used a sentence-pair classifier to classify parallel data as being aligned, or not. Using this approach, we filtered out a large portion of the original, noisy, data and fine-tuned existing large language models on this new data. Our results show that training on the filtered data significantly increases the performance of the models, resulting in improved translations. In particular, our approach outperforms (i) training only on clean data, (ii) training only on filtered data, and (iii) training on the original dataset, consisting of clean and noisy data. This provides additional evidence in favor of prioritizing data quality over quantity, as well as the need for more advanced noise detection

Models	eng→x							fra→x	x→*eng*							x→*fra*	$x \rightarrow$ afr	afr $\rightarrow x$	Avg.
	hau	ibo	lug	swa	tsn	yor	zul	wol	hau	ibo	lug	swa	tsn	yor	zul	wol			
BLEU																			
Clean only	10.7	11.9	4.5	24.3	10.1	**4.2**	6.0	**4.4**	15.7	15.0	12.2	27.5	9.7	8.8	18.5	7.1	9.5	14.3	11.9
Filtered + Clean	**17.7**	**15.3**	**4.6**	**31.5**	**17.8**	3.2	**11.1**	1.5	**22.7**	**20.9**	**15**	**35.2**	**21.2**	**14.2**	**26.8**	**7.6**	**12.8**	**20.4**	**16.6**
CHRF2++																			
Clean only	36.0	34.6	**29.0**	52.2	33.8	**21.8**	36.3	**25.4**	38.0	38.2	33.4	50.4	31.6	29.4	41.6	**28.0**	33.6	36.3	35.0
Filtered + Clean	**43.4**	**38.6**	27.2	**57.7**	**41.9**	19.4	**44.8**	17.9	**45.2**	**44.6**	**35.4**	**57.1**	**43.6**	**35.3**	**49.1**	27.7	**36.4**	**42.2**	**39.4**

Table 8: Performance of the submitted models on the wmt22 test sets as provided by the organizers. We submitted two models. The primary one, denoted *Filtered + Clean*, was trained on the clean bitext as well as the data filtered by ALBERT-xlarge with a threshold of 0.7. The secondary (or contrastive) approach, denoted *Clean only*, was trained only on the clean bitext. The $x \rightarrow$ afr and afr $\rightarrow x$ columns contain the average performance for translations to and from African languages, respectively. *avg* contains the average over all language pairs.

and filtering tools. There are numerous potential avenues for future work; one option is to use a multilingual model as the sentence classifier instead of using a separate model per language, to leverage commonalities between different languages (Adelani et al., 2021b; Conneau et al., 2020). Secondly, a more in-depth study of the effect of the threshold parameter on the final BLEU score would be useful. We would also like to understand the reasons behind the performance by analyzing the filtered data more in depth. Finally, given more computational resources, we will (i) train the classifier for more epochs, using other language models and/or using different quality thresholds, (ii) use longer sentence length than the current 128, (iii) train the translation models on AfroXLMR and ALBERT-base filtered data, and (iv) use the filtering approach on more languages, to evaluate its generalizability. Ultimately, we hope that this filtering approach could lead to the use of cleaner data to train translation models, improving the overall translation quality for low-resourced languages.

Acknowledgements

Idris Abdulmumin acknowledges funding from Google through the Africa Travel Grants. Computations were performed using High Performance Computing infrastructure provided by the Mathematical Sciences Support unit at the University of the Witwatersrand, LIAAD INESC TEC, Portugal, and DFKI GmbH. We acknowledge support from the Machine Learning Group, EISLAB, Luleå University of Science and Technology, Sweden. In addition, we thank the organizers for the compute grant, which allowed us to run some of our experiments. Finally, we thank Dr. Rachel Bawden, and the anonymous reviewers of the 2022 WMT shared task for their helpful feedback. We also acknowledge the useful inputs of Julia Kreutzer and Angela Fan.

References

Ahmed Abdelali, Francisco Guzman, Hassan Sajjad, and Stephan Vogel. 2014. The AMARA corpus: Building parallel language resources for the educational domain. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 1856–1862, Reykjavik, Iceland. European Languages Resources Association (ELRA).

Sadaf Abdul-Rauf, Mark Fishel, Patrik Lambert, Sandra Noubours, and Rico Sennrich. 2012. Extrinsic evaluation of sentence alignment systems. In *Workshop on Creating Cross-language Resources for Disconnected Languages and Styles*, pages 6–10.

Idris Abdulmumin, Bashir Shehu Galadanci, Abubakar Isa, Habeebah Adamu Kakudi, and Ismaila Idris Sinan. 2021. A Hybrid Approach for Improved Low Resource Neural Machine Translation using Monolingual Data. *Engineering Letters*, 29(4):339–350.

Idris Abdulmumin, Bashir Shehu Galadanci, Shamsuddeen Hassan Muhammad, and Garba Aliyu. 2022. Quantity vs. quality of monolingual source data in automatic text translation: Can it be too little if it is too good? In *2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON)*, pages 1–5. IEEE.

David Adelani, Jesujoba Alabi, Angela Fan, Julia Kreutzer, Xiaoyu Shen, Machel Reid, Dana Ruiter, Dietrich Klakow, Peter Nabende, Ernie Chang, Tajuddeen Gwadabe, Freshia Sackey, Bonaventure F. P. Dossou, Chris Emezue, Colin Leong, Michael Beukman, Shamsuddeen Muhammad, Guyo Jarso, Oreen Yousuf, Andre Niyongabo Rubungo, Gilles Hacheme, Eric Peter Wairagala, Muhammad Umair Nasir, Benjamin Ajibade, Tunde Ajayi, Yvonne Gitau, Jade Abbott, Mohamed Ahmed, Millicent Ochieng, Anuoluwapo Aremu, Perez Ogayo, Jonathan Mukiibi, Fatoumata Ouoba Kabore, Godson Kalipe, Derguene Mbaye, Allahsera Auguste Tapo, Victoire Memdjokam Koagne, Edwin

Munkoh-Buabeng, Valencia Wagner, Idris Abdulmumin, Ayodele Awokoya, Happy Buzaaba, Blessing Sibanda, Andiswa Bukula, and Sam Manthalu. 2022a. A few thousand translations go a long way! Leveraging pre-trained models for African news translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3053–3070, Seattle, United States. Association for Computational Linguistics.

David Adelani, Dana Ruiter, Jesujoba Alabi, Damilola Adebonojo, Adesina Ayeni, Mofe Adeyemi, Ayodele Esther Awokoya, and Cristina España-Bonet. 2021a. The effect of domain and diacritics in Yoruba–English neural machine translation. In *Proceedings of Machine Translation Summit XVIII: Research Track*, pages 61–75, Virtual. Association for Machine Translation in the Americas.

David Ifeoluwa Adelani, Jade Z. Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, Stephen Mayhew, Israel Abebe Azime, Shamsuddeen Hassan Muhammad, Chris Chinenye Emezue, Joyce Nakatumba-Nabende, Perez Ogayo, Aremu Anuoluwapo, Catherine Gitau, Derguene Mbaye, Jesujoba Alabi, Seid Muhie Yimam, Tajuddeen Gwadabe, Ignatius Ezeani, Rubungo Andre Niyongabo, Jonathan Mukiibi, Verrah Otiende, Iroro Orife, Davis David, Samba Ngom, Tosin P. Adewumi, Paul Rayson, Mofetoluwa Adeyemi, Gerald Muriuki, Emmanuel Anebi, Chiamaka Chukwuneke, Nkiruka Odu, Eric Peter Wairagala, Samuel Oyerinde, Clemencia Siro, Tobius Saul Bateesa, Temilola Oloyede, Yvonne Wambui, Victor Akinode, Deborah Nabagereka, Maurice Katusiime, Ayodele Awokoya, Mouhamadane Mboup, Dibora Gebreyohannes, Henok Tilaye, Kelechi Nwaike, Degaga Wolde, Abdoulaye Faye, Blessing Sibanda, Orevaoghene Ahia, Bonaventure F. P. Dossou, Kelechi Ogueji, Thierno Ibrahima Diop, Abdoulaye Diallo, Adewale Akinfaderin, Tendai Marengereke, and Salomey Osei. 2021b. Masakhaner: Named entity recognition for african languages. *Trans. Assoc. Comput. Linguistics*, 9:1116–1131.

David Ifeoluwa Adelani, Md Mahfuz Ibn Alam, Antonios Anastasopoulos, Akshita Bhagia, Marta Costa-Jussá, Jesse Dodge, Fahim Faisal, Christian Federmann, Natalia Fedorova, Francisco Guzmán, Sergey Koshelev, Jean Maillard, Vukosi Marivate, Jonathan Mbuya, Safiyyah Saleem, and Holger Schwenk. 2022b. Findings of the WMT 2022 shared task on large-scale machine translation evaluation for african languages. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Michael Adjeisah, Guohua Liu, Douglas Omwenga Nyabuga, Richard Nuetey Nortey, and Jinling Song. 2021. Pseudotext injection and advance filtering of low-resource corpus for neural machine translation. *Comput. Intell. Neurosci.*, 2021:6682385:1–6682385:10.

Jesujoba Alabi, Kwabena Amponsah-Kaakyire, David Adelani, and Cristina España-Bonet. 2020. Massive vs. curated embeddings for low-resourced languages: the case of Yorùbá and Twi. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2754–2762, Marseille, France. European Language Resources Association.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pretrained language models to African languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4336–4349, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Karunesh Kumar Arora, Geetam S Tomar, and Shyam S Agrawal. 2021. Studying the role of data quality on statistical and neural machine translation. In *2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*, pages 199–204. IEEE.

Mikel Artetxe and Holger Schwenk. 2019a. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019b. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Trans. Assoc. Comput. Linguistics*, 7:597–610.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 355–362. ACL.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Ondrej Bojar and Ales Tamchyna. 2011. Improving translation model by monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine*

Translation, WMT@EMNLP 2011, Edinburgh, Scotland, UK, July 30-31, 2011, pages 330–336. Association for Computational Linguistics.

Christos Christodouloupoulos and Mark Steedman. 2015. A massively parallel corpus: the Bible in 100 languages. *Language Resources and Evaluation*, 49(2):375–395.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Sourav Dutta, Jesujoba Alabi, Saptarashmi Bandyopadhyay, Dana Ruiter, and Josef van Genabith. 2020. UdS-DFKI@WMT20: Unsupervised MT and very low resource supervised MT for German-Upper Sorbian. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1092–1098, Online. Association for Computational Linguistics.

Sauleh Eetemadi, William Lewis, Kristina Toutanova, and Hayder Radha. 2015. Survey of data-selection methods in statistical machine translation. *Mach. Transl.*, 29(3-4):189–223.

Sauleh Eetemadi and Kristina Toutanova. 2015. Detecting translation direction: A cross-domain study. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 103–109, Denver, Colorado. Association for Computational Linguistics.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 5960–5969, Online. Association for Computational Linguistics.

Ahmed El-Kishky, Adi Renduchintala, James Cross, Francisco Guzmán, and Philipp Koehn. 2021. XLEnt: Mining cross-lingual entities with lexical-semantic-phonetic word alignment. In *Preprint*, Online.

Miquel Esplà, Mikel Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. 2019. ParaCrawl: Web-scale parallel corpora for the languages of the EU. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 118–119, Dublin, Ireland. European Association for Machine Translation.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(1).

Guillem Gascó, Martha Alicia Rocha, Germán Sanchis-Trilles, Jesús Andrés-Ferrer, and Francisco Casacuberta. 2012. Does more data always yield better translations? In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics - EACL 2012*, pages 152–161. Association for Computational Linguistics.

Ona de Gibert, Ksenia Kharitonova, Blanca Calvo Figueras, Jordi Armengol-Estap'e, and Maite Melero. 2022. Quality versus quantity: Building catalan-english mt resources. In *Proceedings of SIGUL2022 @LREC2022*, pages 59–69, Marseille. European Language Resources Association (ELRA).

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindrich Helcl, and Alexandra Birch. 2022. Survey of low-resource machine translation. *Comput. Linguistics*, 48(3):673–732.

Tahmid Hasan, Abhik Bhattacharjee, Kazi Samin, Masum Hasan, Madhusudan Basak, M. Sohel Rahman, and Rifat Shahriyar. 2020. Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for bengali-english machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2612–2623. Association for Computational Linguistics.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Aizhan Imankulova, Takayuki Sato, and Mamoru Komachi. 2017. Improving low-resource neural machine translation with filtered pseudo-parallel corpus. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, pages 70–78, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 888–895. Association for Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzmán. 2020. Findings of the WMT 2020 shared task on parallel corpus filtering and alignment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 726–742, Online. Association for Computational Linguistics.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 54–72, Florence, Italy. Association for Computational Linguistics.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018. Findings of the WMT 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 726–739, Belgium, Brussels. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroro Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2022. Quality at a Glance: An Audit of Web-Crawled Multilingual Datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Patrik Lambert, Holger Schwenk, Christophe Servan, and Sadaf Abdul-Rauf. 2011. Investigations on translation model adaptation using monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation, WMT@EMNLP 2011, Edinburgh, Scotland, UK, July 30-31, 2011*, pages 284–293. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Wilhelmina Nekoto, Vukosi Marivate, Tshinondiwa Matsila, Timi E. Fasubaa, Taiwo Fagbohungbe, Solomon Oluwole Akinola, Shamsuddeen Hassan Muhammad, Salomon Kabongo Kabenamualu, Salomey Osei, Freshia Sackey, Rubungo Andre Niyongabo, Ricky Macharm, Perez Ogayo, Orevaoghene Ahia, Musie Meressa Berhe, Mofetoluwa Adeyemi, Masabata Mokgesi-Selinga, Lawrence Okegbemi, Laura Martinus, Kolawole Tajudeen, Kevin Degila, Kelechi Ogueji, Kathleen Siminyu, Julia Kreutzer, Jason Webster, Jamiil Toure Ali, Jade Z. Abbott, Iroro Orife, Ignatius Ezeani, Idris Abdulkabir Dangana, Herman Kamper, Hady Elsahar, Goodness Duru, Ghollah Kioko, Espoir Murhabazi, Elan Van Biljon, Daniel Whitenack, Christopher Onyefuluchi, Chris Chinenye Emezue, Bonaventure F. P. Dossou, Blessing Sibanda, Blessing Itoro Bassey, Ayodele Olabiyi, Arshath Ramkilowan, Alp Öktem, Adewale Akinfaderin, and Abdallah Bashir. 2020. Participatory research for low-resourced machine translation: A case study in african languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 2144–2160. Association for Computational Linguistics.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. Association for Computational Linguistics.

Vincent Nguyen, Sarvnaz Karimi, and Zhenchang Xing. 2021. Combining shallow and deep representations for text-pair classification. In *Proceedings of the The 19th Annual Workshop of the Australasian Language Technology Association*, pages 68–78, Online. Australasian Language Technology Association.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia-Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Michael Przystupa and Muhammad Abdul-Mageed. 2019. Neural machine translation of low-resource and similar languages with backtranslation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 224–235, Florence, Italy. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Philip Resnik. 1999. Mining the web for bilingual text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 527–534, College Park, Maryland, USA. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Svetlana Tchistiakova, Jesujoba Alabi, Koel Dutta Chowdhury, Sourav Dutta, and Dana Ruiter. 2021. EdinSaar@WMT21: North-germanic low-resource multilingual NMT. In *Proceedings of the Sixth Conference on Machine Translation*, pages 368–375, Online. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey. European Languages Resources Association (ELRA).

Nicola Ueffing. 2006. Using Monolingual Source-Language Data to Improve MT Performance. In *International Workshop on Spoken Language Translation*, pages 174–181, Kyoto, Japan.

Guanghao Xu, Youngjoong Ko, and Jungyun Seo. 2019. Improving neural machine translation by filtering synthetic parallel data. *Entropy*, 21(12):1213.

A Appendix - Data Sources

Datasets used in this project and their sources, as listed in Table 9: MAFAND-MT, wmt22_african, LAVA Corpus,[9] XLEnt, Tanzil, WikiMatrix, CCAligned, CCMatrix, GlobalVoices,[10,11] ParaCrawl,[12] GNOME,[13] tico-19,[14] ELRC_2922,[15] EUbookshop,[16] KDE4,[17] TED2020, Tatoeba,[18] Ubuntu,[19] bible-uedin, wikimedia,[20] QED, MultiCCAligned and Mozilla-I10n.

[9] https://drive.google.com/drive/folders/179AkJ0P3fZMFS0rIyEBBDZ-WICs2wpWU

[10] https://casmacat.eu/corpus/global-voices.html

[11] https://globalvoices.org/

[12] https://paracrawl.eu/

[13] https://l10n.gnome.org/

[14] https://tico-19.github.io/index.html

[15] https://elrc-share.eu/repository/browse/covid-19-health-wikipedia-dataset-multilingual-53-en-x-language-pairs/fe23e2c28c8311ea913100155d0267066f62c6b30ac0429f8d497df0abd2ef72/

[16] http://bookshop.europa.eu

[17] http://www.lt-innovate.org/lt-observe/resources/kde4-kde4-localization-files-v2

[18] https://tatoeba.org/en/

[19] https://translations.launchpad.net/

[20] https://dumps.wikimedia.org/other/contenttranslation/

Data	en							fr
	hau	ibo	lug	swa	tsn	yor	zul	wol
True Parallel								
MAFAND-MT	3,098	6,998	4,075	30,782	2,100	6,644	3,500	3,360
Tanzil	128,376	-	-	138,253	-	-	-	-
GlobalVoices	-	-	-	32,307	-	137	-	-
tico-19	3,071	-	3,071	3,071	-	-	3,071	-
ELRC_2922	-	-	-	607	-	-	-	-
EUbookshop	-	-	-	18	-	-	-	-
Tatoeba	57	22	3	395	31	37	70	67
bible-uedin	-	-	-	-	-	-	15,907	7,918
QED	124	12	740	18,192	-	52	1,624	66
Mozilla-I10n	4,952	4,172	5,931	7,798	-	4,095	-	7,041
Total (TP)	139,678	11,204	13,820	231,423	2,131	10,965	24,172	18,452
Automatcally Aligned								
WMT22 African	2,309,758	172,973	3,450,573	23,358,739	5,931,529	1,455,571	3,862,020	189,659
WebCrawl Afr.	16,950	3,372	10,809	193,518	77,976	18,924	152,724	-
LAVA Corpus	-	-	20,993	371,864	-	-	-	-
WikiMatrix	-	-	-	51,387	-	-	-	-
CCAligned	339,178	148,147	14,702	2,044,993	71,254	175,193	126,103	–
CCMatrix	5,861,080	80,385	-	5,756,664	-	-	-	-
ParaCrawl	-	-	-	132,521	-	-	-	-
GNOME	5,466	23,767	4,578	40	-	10,234	44,605	-
KDE4	1,493	-	-	-	-	-	-	-
TED2020	27	210	-	9,745	-	-	-	-
XLEnt	436,602	69,820	1,054	871,902	4,781	51,173	28,394	4,082
Ubuntu	242	635	637	986	-	141	4,718	220
wikimedia	23,385	12,279	1,315	3,765	969	8,521	1,226	679
MultiCCAligned	-	-	-	-	-	-	-	24,256
Total (AA)	8,994,181	511,588	3,504,661	32,796,124	6,086,509	1,719,757	4,219,790	218,896
Total (ALL)	9,133,859	522,792	3,518,481	33,027,547	6,088,640	1,730,722	4,243,962	237,348

Table 9: Training Data Used — TP=True Parallel; AA=Automatically Aligned

Language Adapters for Large-Scale MT: The GMU System for the WMT 2022 Large-Scale Machine Translation Evaluation for African Languages Shared Task

Md Mahfuz Ibn Alam Antonios Anastasopoulos
Department of Computer Science, George Mason University
{malam21,antonis}@gmu.edu

Abstract

This report describes GMU's machine translation systems for the WMT22 shared task on large-scale machine translation evaluation for African languages (Adelani et al., 2022b). We participated in the constrained translation track where only the data listed on the shared task page were allowed, including submissions accepted to the Data track. Our approach uses models initialized with DeltaLM, a generic pre-trained multilingual encoder-decoder model, and fine-tuned correspondingly with the allowed data sources. Our best submission incorporates language family and language-specific adapter units; ranking ranked second under the constrained setting.

1 Introduction

There has traditionally been a significant concentration of machine translation research on a few languages - usually Indo-European (Blasi et al., 2022). Data scarcity has hindered the progress of many languages, many with millions of speakers (Joshi et al., 2020). The shared task and our submission aim to reverse the trend by focusing on low-resource African languages that have been traditionally ignored by mainstream research.

Our submission leverages different approaches to produce a multilingual MT system that can handle all 26 languages covered by the shared task:

- All data available under the constrained setting,
- Delta-LM (Ma et al., 2021), a pre-trained multilingual encoder-decoder model,
- adapter units (Houlsby et al., 2019) are designed to adapt the multilingual model to specific language pairs, and
- phylogeny-inspired organization of the adapters (Faisal and Anastasopoulos, 2022), which allows for information sharing across similar (related) languages.

We expand on each of these components in our system description and the related work section.

Our DeltaLM model was fine-tuned in the first step using parallel data collected from all 26 languages. After fine-tuning the previous model, we adapter-tune the language-specific adapters. Our third step is to adapter-tune the family-specific and sub-family-specific adapters based on the previous adapter-tune model. We submit the second and third models as our submissions to the shared task.

2 Data

Data Sources We use bilingual data from multiple sources. Our main source was the OPUS-100[1] website and Shared Task[2] website. The datasets are:

- ELRC, KDE4, OpenSubtitles, GlobalVoices, Tanzil, EUbookshop, Europarl, infopankki, memat, Tatoeba, Wikimedia) (Tiedemann, 2012),
- MultiCCAligned, CCAligned (El-Kishky et al., 2020),
- WikiMatrix (Schwenk et al., 2019a),
- QED (Abdelali et al., 2014), bible (Christodouloupoulos and Steedman, 2015),
- CCMatrix (Schwenk et al., 2019b),
- TED (Reimers and Gurevych, 2020),
- ParaCrawl (Bañón et al., 2020),
- NLLB Crawled Data (NLLB Team et al., 2022),
- LAVA corpus,[3]
- MAFAND-MT[4] (Adelani et al., 2022a),

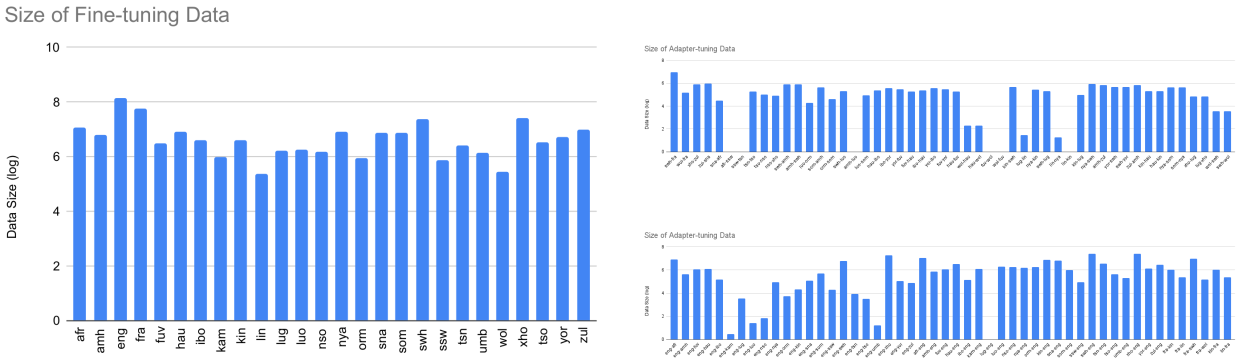
[1] https://opus.nlpl.eu/

[2] https://www.statmt.org/wmt22/large-scale-multilingual-translation-task.html

[3] https://drive.google.com/drive/folders/179AkJ0P3fZMFS0rIyEBBDZ-WICs2wpWU

[4] https://github.com/masakhane-io/

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1015 - 1033
December 7-8, 2022 ©2022 Association for Computational Linguistics



(a) Bilingual data statistics of the 26 languages for fine-tuning. The columns indicate the size of data for each language in comparison to the remaining 25 languages.

(b) Data-set statistics of the bilingual data of the 100 language pairs for adapter-tuning.

Figure 1: Data statistics for fine-tuning (left) and adapter-tuning (right). Training data size is logarithmically transformed (base 10) for better visualization.

- WebCrawl African[5] (Vegi et al., 2022),
- KenTrans[6] (Wanjawa et al., 2022).

Figure 1(a) shows the data-statistics of the bilingual data for 26 languages. We use these data to fine-tune the DeltaLM model at first. Figure 1(b) shows the data statistics of the bilingual data for 100 language pairs. We use these data to adapter-tune the fine-tuned model at first for language adapters and then for family and sub-family adapters.

2.1 Data Pre-Processing

Filtering We removed sentences longer than 768 words and shorter than five words. We removed sentences where the whole sentence was made of punctuation. After that, we removed duplicate sentence pairs from the whole data set.

Tokenization After data filtering, we used the SentencePiece model (Kudo and Richardson, 2018) to tokenize all raw training and validation datasets. We keep the SentencePiece model consistent with the one used for DeltaLM.

Use in Training We shuffled the whole training dataset before launching the fine-tuning of multilingual models. Our multilingual model was then fine-tuned on the entire dataset; note that the dataset is potentially noisy as we have not removed any sentence pairs which have potentially incorrect language identification or character encoding. Each source sentence was prefixed with a tag to indicate the target language. For example, the English source sentence `"I love MT"` would be changed to `"<am> I love MT"` to translate into Amharic.

3 Model and Training

3.1 Initialization with DeltaLM

We have based all our experiments on the DeltaLM_large architecture, which consists of 24 Transformer encoder layers and 12 interleaved decoder layers with embedding sizes of 1024, dropouts of 0.1, feed-forward networks of 4096, and attention heads of 16. We directly initialize our model with the publicly available DeltaLM_large checkpoint.

3.2 Multilingual Fine-tuning

Given training data as bi-text corpora $D_b = \{D_b^1, D_b^2, ..., D_b^n\}$, where n is the number of different translation directions. For 26 languages n is 625. We mix all corpora of all directions and shuffle the whole data $D_b^{1...n}$. Then we optimize the model's parameters θ using the standard NLL objective:

$$L_{MT} = E_{\mathbf{x},\mathbf{y} \epsilon D_b^{1...n}}[-logP(\mathbf{y}|\mathbf{x};\theta)]$$

Where $\mathbf{x}, \mathbf{y}$ denotes a sentence pair. $L_M T$ is the translation objective for the multilingual model. We refer to this model as "Fine-Tune" for the remainder of the paper.

lafand-mt/tree/main/data/text_files

[5] https://github.com/pavanpankaj/Web-Crawl-African

[6] https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/NOAT0W

3.3 Multilingual Adapter-tuning

Adapter Units Between the layers of the pre-trained network, we have added lightweight adapter layers and fine-tuned them using the same corpus as above. In each adapter, an up projection to the starting dimension follows a down projection to a bottleneck dimension (Philip et al., 2020). The bottleneck keeps the number of parameters of the adapter module at a limit. A residual connection coupled with a near-identity initialization enables a pass-through and allows us to maintain the parent model's performance while training the adapter units.

The training data is also the bi-text corpora $D_b = \{D_b^1, D_b^2, ..., D_b^{100}\}$ for the 100 language directions specified by the shared task evaluation schema. We trained the multilingual model as before, but now training only the parameters of the adapters $\theta_{Adapter}$:

$$L_{MT} = \sum_{i=1}^{100} E_{\mathbf{x},\mathbf{y}\in D_b^i}[-logP(\mathbf{x}|\mathbf{y};\theta_{Adapter})]$$

where $\theta_{Adapter}$ are the parameters of the adapters only; i denotes the language direction. In this stage, we add language-specific adapters as shown in Figure 2(a) to every layer of the encoder and decoder. The adapters of the same language on the encoder and decoder side do not share parameters. We refer to this model as "Language-Tune".

Family-specific Adapter In this stage, we add family-specific and genus-specific adapters along with language-specific adapters as a stack, as shown in Figure 2(b), to every layer of the encoder and decoder. The adapters on the encoder and decoder side of the same language, family, and sub-family do not share parameters. But for languages that belong in the same family or genus (group), their family and genus adapters are shared. For example, the Afro-Asiatic family adapter is shared between Hausa, Amharic, Oromo, and Somali, and Oromo and Somali also share the Cushitic adapter. The training data and optimization objective is the same as above.

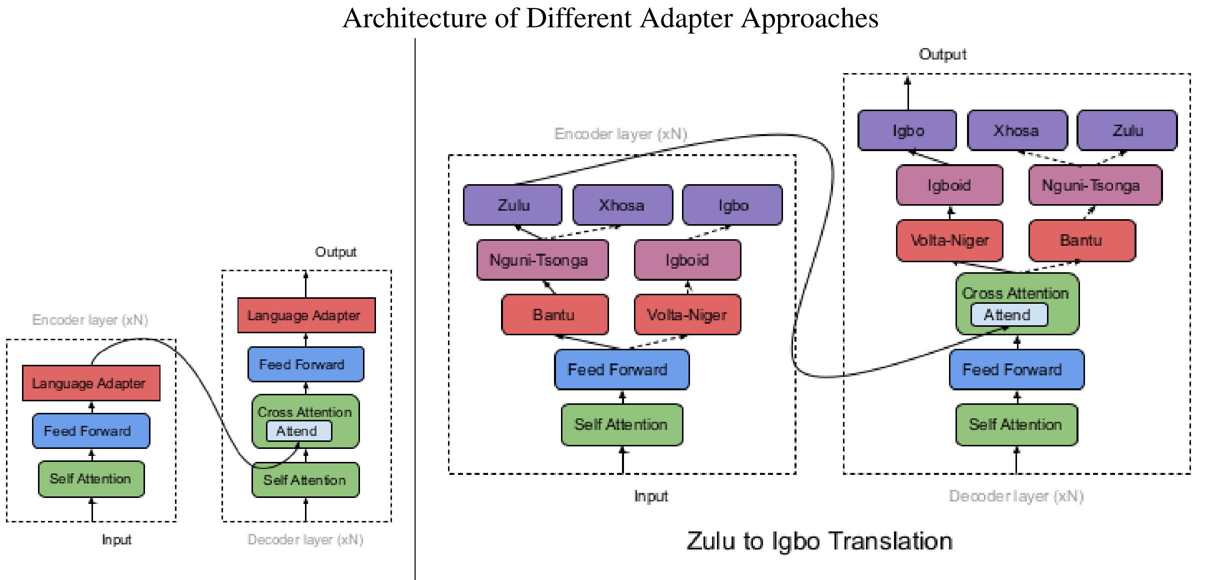
Table 1 shows the phylogeny-informed tree-hierarchy of all 26 languages. On the encoder side, only adapters associated with the source language are active for a specific language direction. On the decoder side, the adapters associated with the target language get active. For example, when training (or translating) from Nigerian Fulfulde to Xhosa, the `Senegambian`, `Fula`, and `Nigerian Fulfulde` adapters will be used on the encoder side, and the `Bantu`, `Nguni-Tsonga`, and `Xhosa` adapters will be used on the decoder side. The resulting model will be referred to as "Family-Tune" for the rest of the paper.

Family	Genus (Group)	Language
Indo-European	Germanic	English Afrikaans
	Romance	French
Afro-Asiatic	Hausa	Hausa
	Amharic	Amharic
	Cushitic	Oromo
	Cushitic	Somali
Nilo-Saharan	Luo	Luo
Senegambian	Wolof	Wolof
	Fula	Nigerian Fulfulde
Volta-Niger	Igboid	Igbo
	Yoruboid	Yoruba
Bantu	Bangi	Lingala
	Shona	Shona
	Nyasa	Chichewa
	Umbundu	Umbundu
	Sotho-Tswana	Tswana
		Northern Sotho
	Nguni-Tsonga	Zulu
		Xhosa
		Swati
		Xitsonga
	Northeast-Bantu	Kamba
		Swahili
		Kinyarwanda
		Luganda

Table 1: The phylogeny-informed language tree hierarchy that we impose on our language adapters.

3.4 Training Details

Fine-Tuning We train multilingual models with the Adam optimizer (Kingma and Ba, 2014) ($\beta1 = 0.9$, $\beta2 = 0.98$). The learning rate is set as 1e-4 with a warm-up step of 4000. The models are trained with label smoothing with a ratio of 0.1. All experiments are conducted on 4 A100 GPUs. The batch size is 1536 tokens per GPU, and the model is updated every 4 (for 4 A100 GPUs) steps



(a) Language-specific Adapter

(b) Family, Sub-family and Language-specific Adapter

Figure 2: Current practice uses language-specific adapters between layers (a). In order to incorporate linguistic information into our models, we impose phylogenetic tree hierarchies based on phylogeny, as in (b), where the solid line shows the path the model has to take for Zulu to Igbo translation, and dotted lines show other possible paths for different language pairs.

to simulate a larger batch size. We have kept the max source and target positions as 512 and have skipped any inputs that have invalid sizes.

Adapter-Tuning We use the same parameters as above. As we do not use the whole dataset to train but data of each language direction, we set the warm-up step as 1000. We train the model for a maximum of 5 epochs or a maximum of 20000 updates (whichever comes first). The dimension of the bottleneck layer of the adapter on both the encoder and decoder sides is set to 64.

Language-Specific We add language adapters to DeltaLM and train only the adapters and keep all other parameters frozen.

Family-Specific We add family and sub-family adapters to DeltaLM where language adapters have already been inserted. We train only the family and sub-family adapters and keep all other parameters frozen, including the language adapters.

4 Evaluation Results

We use the dev and the hidden test set of the FLORES200 (Guzmán et al., 2019; Goyal et al., 2021; NLLB Team et al., 2022) benchmark as our validation set and test set respectively. A beam search strategy with a beam size of 5 is used during inference in order to generate target sentences. Based on the loss on the validation set, we select the best checkpoint for evaluation. We report BLEU, CHRF++, and SentencePiece-based BLEU using spBLEU scores.

Our model using language-specific adapters significantly outperforms the fine-tuning model. Table 2 shows that the model with language-specific adapters outperforms the fine-tuning model on average for all directions from 0.2 to 0.6 BLEU points. Our work solidifies the argument made in previous work that some language-specific elements help the model to better model each language.

Our model with family-specific adapters does not seem to outperform the language-specific adapters on average. But we do obtain some gains for **Avg**$_{X\rightarrow eng}$ and **Avg**$_{fra\rightarrow Y}$. Going deeper to the results, we do find significant gains for some individual language pairs: for instance, for Tswana-English (tsn-eng) we obtain a 1.0 BLEU point gain, and for English-Hausa (eng-hau) this model is better by 1.2 BLEU points.

Table 3 shows that our model with language-specific adapters also achieves better results than the fine-tuning model for different regions of African to African language pairs. We were able to gain BLEU points from 0.1 to 0.25 on average. For family-specific adapters, we see some gains for some regions like Nigeria and for translating between regions.

Metrics	Models	**Avg**$_{all}$	**Avg**$_{X\rightarrow eng}$	**Avg**$_{eng\rightarrow X}$	**Avg**$_{African\rightarrow African}$	**Avg**$_{Y\rightarrow fra}$	**Avg**$_{fra\rightarrow Y}$
BLEU	Fine-Tune	13.00	25.44	11.62	7.57	20.28	10.03
	Language-Tune	**13.28**	25.83	**12.00**	**7.70**	**20.83**	10.53
	Family-Tune	13.28	**25.88**	11.98	7.68	20.73	**10.75**
CHRF++	Fine-Tune	34.80	45.82	34.52	29.56	41.55	31.85
	Language-Tune	**35.42**	46.50	**35.33**	**29.94**	**42.45**	33.58
	Family-Tune	35.42	**46.55**	35.30	29.92	42.30	**34.03**
spBLEU	Fine-Tune	15.85	27.45	14.78	10.64	23.80	12.55
	Language-Tune	**16.23**	27.97	**15.24**	**10.85**	**24.30**	13.55
	Family-Tune	16.20	**28.00**	15.12	10.82	24.28	**13.65**

Table 2: Evaluation results of Constrained Track for our methods of 100 language directions on the hidden test set of the FLORES-200 benchmark. **Avg**$_{X\rightarrow eng}$ denotes the average score of directions between other languages and English. **Avg**$_{eng\rightarrow X}$ denotes the average score of directions between English and other languages. **Avg**$_{African\rightarrow African}$ denotes the average score of directions between African languages to other African languages. **Avg**$_{Y\rightarrow fra}$ denotes the average score of directions between other languages and French. **Avg**$_{fra\rightarrow Y}$ denotes the average score of directions between French and other languages. **Avg**$_{all}$ denotes the average result of all translation directions.

Tables 5, 6, and 7 show the complete results on all 100 language pairs tested on devtest, hidden test and on the TICO-19 (Anastasopoulos et al., 2020) dataset.

Discussion on Pre-training Membership Among the 24 African languages, only 7 of them (Afrikaans, Amharic, Hausa, Oromo, Somali, Swahili, and Xhosa) were used in the pre-training of the DeltaLM model. As previous work has shown (Muller et al., 2021), models tend to perform worse for languages not included in pre-training. Nevertheless, our model is still competitive; we attribute this to the fact that we have used any dataset that we could get our hands on from the OPUS website discarding the fact that these data may be noisy or may have high domain mismatch.

Table 4 shows the result between languages present in the pre-training of DeltaLM vs languages not present. For all averages, we see the same trend as adapter-tuning is better than the fine-tuned model. Between non-present languages (npl) and present languages (pl) we see **Avg**$_{npl}$, **Avg**$_{pl}$, **Avg**$_{npl-source}$ and **Avg**$_{pl-source}$ shows the same pattern where the present languages have higher scores than the non-present languages. But we see the opposite pattern for **Avg**$_{npl-target}$ and **Avg**$_{pl-target}$ where the present languages have lower average.

Limitations One glaring limitation of our approach is that it is not making use of the potentially large amounts of monolingual data in the languages, e.g. through back-translation (Sennrich et al., 2016). In our training, we have not used any monolingual data at all. Monolingual data are more available than parallel data and are less noisy. We could have used monolingual data to pre-train the DeltaLM with the span corruption objective. We could then use that pre-trained model as our base model to fine-tune using the parallel data. We could also do iterative back-translation using the monolingual data to create synthetic parallel data and train the model with these data along with the real parallel data. This approach has proven to be effective for low-resourced settings before, and we will further explore it in future work.

In addition, our phylogeny-inspired adaptors follow a pre-defined path along the trees. This is perhaps too rigid, especially for communities that use a lot of code-switching, or for creole languages and pidgins that are the result of language contact. In future work, we will explore ways to *learn* the path through the tree, or allow for soft sharing of parameters through attention or mixture of experts units.

5 Related Work

Multilingual neural machine translation (Dong et al., 2015; Johnson et al., 2016; Arivazhagan et al., 2019; Dabre et al., 2020; Philip et al., 2020; Lin et al., 2021) is now the de facto architecture because of its ability to produce translations between

Metrics	Models	$\textbf{Avg}_{south-east}$	$\textbf{Avg}_{horn}$	$\textbf{Avg}_{nigeria}$	$\textbf{Avg}_{central}$	$\textbf{Avg}_{among-region}$
BLEU	Fine-Tune	12.35	6.31	4.32	9.23	7.36
	Language-Tune	**12.48**	**6.55**	4.39	**9.35**	7.50
	Family-Tune	12.34	6.50	**4.44**	9.31	**7.53**
CHRF++	Fine-Tune	40.80	28.20	18.98	33.80	30.73
	Language-Tune	**41.08**	**28.83**	19.13	**34.15**	31.26
	Family-Tune	40.89	28.76	**19.28**	34.05	**31.28**
spBLEU	Fine-Tune	17.34	10.53	5.32	11.56	10.98
	Language-Tune	**17.51**	**10.85**	5.36	**11.76**	**11.29**
	Family-Tune	17.34	10.83	**5.47**	11.68	11.27

Table 3: Evaluation results of Constrained Track for our methods of 38 African to African language directions on the hidden test set of the FLORES-200 benchmark.

multiple languages. This is because there are thousands of languages worldwide, and if we were to make bilingual models, we would need thousands of models to represent all the languages. This is not ideal because it is neither scalable nor adaptable. Various research tries to improve the performance of multilingual translation models. Either through various training methods (Aharoni et al., 2019; Wang et al., 2020), model structures (Wang et al., 2018; Gong et al., 2021; Zhang et al., 2021), or data augmentation (Tan et al., 2019; Pan et al., 2021). The M2M model (Fan et al., 2020) utilizes large-scale data derived from the web and explores the techniques for enlarging the model and effectively training it.

Multilingual pre-trained language models like mBART (Liu et al., 2020) which pre-trains a multilingual model with the multilingual denoising objective, have proven to be effective in improving multilingual machine translation. These pre-trained models also have drawbacks, like adapting to new languages not seen during pre-training.

Adapters (Houlsby et al., 2019) are designed to adapt a large pre-trained model to a downstream task with lightweight residual layers (Rebuffi et al., 2018) that are inserted into each layer of the model. As part of machine translation, Bapna et al. (2019) proposed bilingual adapters to improve pre-trained multilingual machine translation models or to adapt them to domains. Philip et al. (2020) designed language-specific adapters to improve zero-shot machine translation. Finally, Stickland et al. (2020) use language-agnostic task adapters for fine-tuning BART and mBART to bilingual and multilingual MT. Faisal and Anastasopoulos (2022) imposes a phylogeny-informed tree hierarchy over adapters, leading to improved zero-shot performance for languages unseen during pre-training in tasks like dependency parsing. Our work, in contrast to previous ones, uses the family-specific and genus-specific adapters on top of language-specific adapters as a stack for encoder-decoder models and for generation tasks like machine translation, to leverage the idea that languages in the same family should have similar traits. This may aid languages with very little parallel corpora which may be related to other languages with more resources.

6 Conclusion

This paper describes GMU's submission to the large-scale machine translation for African languages of the WMT22 shared task. Here we explore if pre-trained models can be useful even for languages on which they have not been pre-trained. Our multilingual adapter-tuning translation model, built on DeltaLM, achieves substantial improvements over simply fine-tuning DeltaLM. We further try to enhance the model performance with adapter-tuning using phylogeny information. As a result, our submitted systems rank third on the data-constrained track.

Acknowledgements

This work is generously supported by NSF Award IIS-2125466. We thank the Office of Research Computing of GMU for giving us access to the Argo and Hopper clusters for training on A100 GPUs.

Metrics	Models	Avg$_{npl}$	Avg$_{pl}$	Avg$_{npl-source}$	Avg$_{pl-source}$	Avg$_{npl-target}$	Avg$_{pl-target}$
Bleu	Fine-Tune	12.88	13.12	12.47	14.49	13.70	11.10
	Language-Tune	**13.14**	13.41	12.78	**14.69**	**13.95**	11.46
	Family-Tune	13.13	**13.43**	**12.79**	14.67	13.93	**11.51**
CHRF++	Fine-Tune	34.06	35.57	34.18	36.57	35.07	34.06
	Language-Tune	**34.66**	36.20	**34.86**	37.00	35.63	34.83
	Family-Tune	34.64	**36.24**	34.85	**37.07**	**35.64**	**34.84**
spBLEU	Fine-Tune	15.23	16.50	15.14	17.87	16.18	14.98
	Language-Tune	**15.61**	**16.87**	**15.55**	**18.17**	**16.54**	15.39
	Family-Tune	15.57	16.85	15.53	18.12	16.49	**15.41**

Table 4: Evaluation results of Constrained Track for our methods of languages present in the pre-training of DeltaLM vs languages not present. **Avg**$_{npl}$ denotes the average score of language directions where no language was present in the pre-training of DeltaLM. **Avg**$_{pl}$ denotes the average score of language directions where at least one language was present in the pre-training of DeltaLM. **Avg**$_{npl-source}$ denotes the average score of language directions where the source language was not present in the pre-training of DeltaLM. **Avg**$_{pl-source}$ denotes the average score of language directions where the source language was present in the pre-training of DeltaLM. **Avg**$_{npl-target}$ denotes the average score of language directions where the target language was not present in the pre-training of DeltaLM. **Avg**$_{pl-target}$ denotes the average score of language directions where the target language was present in the pre-training of DeltaLM.

References

Ahmed Abdelali, Francisco Guzman, Hassan Sajjad, and Stephan Vogel. 2014. The AMARA corpus: Building parallel language resources for the educational domain. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1856–1862, Reykjavik, Iceland. European Language Resources Association (ELRA).

David Adelani, Jesujoba Alabi, Angela Fan, Julia Kreutzer, Xiaoyu Shen, Machel Reid, Dana Ruiter, Dietrich Klakow, Peter Nabende, Ernie Chang, Tajuddeen Gwadabe, Freshia Sackey, Bonaventure F. P. Dossou, Chris Emezue, Colin Leong, Michael Beukman, Shamsuddeen Muhammad, Guyo Jarso, Oreen Yousuf, Andre Niyongabo Rubungo, Gilles Hacheme, Eric Peter Wairagala, Muhammad Umair Nasir, Benjamin Ajibade, Tunde Ajayi, Yvonne Gitau, Jade Abbott, Mohamed Ahmed, Millicent Ochieng, Anuoluwapo Aremu, Perez Ogayo, Jonathan Mukiibi, Fatoumata Ouoba Kabore, Godson Kalipe, Derguene Mbaye, Allahsera Auguste Tapo, Victoire Memdjokam Koagne, Edwin Munkoh-Buabeng, Valencia Wagner, Idris Abdulmumin, Ayodele Awokoya, Happy Buzaaba, Blessing Sibanda, Andiswa Bukula, and Sam Manthalu. 2022a. A few thousand translations go a long way! leveraging pre-trained models for African news translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3053–3070, Seattle, United States. Association for Computational Linguistics.

David Ifeoluwa Adelani, Md Mahfuz Ibn Alam, Antonios Anastasopoulos, Akshita Bhagia, Marta Costa-Jussá, Jesse Dodge, Fahim Faisal, Christian Federmann, Natalia Fedorova, Francisco Guzmán, Sergey Koshelev, Jean Maillard, Vukosi Marivate, Jonathan Mbuya, Safiyyah Saleem, and Holger Schwenk. 2022b. Findings of the WMT 2022 shared task on large-scale machine translation evaluation for african languages. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. *CoRR*, abs/1903.00089.

Antonios Anastasopoulos, Alessandro Cattelan, Zi-Yi Dou, Marcello Federico, Christian Federmann, Dmitriy Genzel, Franscisco Guzmán, Junjie Hu, Macduff Hughes, Philipp Koehn, Rosie Lazar, Will Lewis, Graham Neubig, Mengmeng Niu, Alp Öktem, Eric Paquin, Grace Tang, and Sylwia Tur. 2020. TICO-19: the translation initiative for COvid-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, Online. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Ankur Bapna, Naveen Arivazhagan, and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. *CoRR*, abs/1909.08478.

Damian Blasi, Antonios Anastasopoulos, and Graham Neubig. 2022. Systematic inequalities in language technology performance across the world's languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5486–5505, Dublin, Ireland. Association for Computational Linguistics.

Christos Christodouloupoulos and Mark Steedman. 2015. A massively parallel corpus: the bible in 100 languages. *Language Resources and Evaluation*, 49(2):375–395.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A comprehensive survey of multilingual neural machine translation. *CoRR*, abs/2001.01115.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 5960–5969, Online. Association for Computational Linguistics.

Fahim Faisal and Antonios Anastasopoulos. 2022. Phylogeny-inspired adaptation of multilingual models to new languages.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *CoRR*, abs/2010.11125.

Hongyu Gong, Xian Li, and Dmitriy Genzel. 2021. Adaptive sparse transformer for multilingual translation.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2021. The FLORES-101 evaluation benchmark for low-resource and multilingual machine translation. *CoRR*, abs/2106.03193.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Miguel Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. Two new evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. *CoRR*, abs/1902.01382.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. *CoRR*, abs/1902.00751.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's multilingual neural machine translation system: Enabling zero-shot translation. *CoRR*, abs/1611.04558.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, abs/1808.06226.

Zehui Lin, Liwei Wu, Mingxuan Wang, and Lei Li. 2021. Learning language specific sub-network for multilingual machine translation. *CoRR*, abs/2105.09259.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *CoRR*, abs/2001.08210.

Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, and Furu Wei. 2021. Deltalm: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders. *CoRR*, abs/2106.13736.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. *CoRR*, abs/2105.09501.

Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual adapters for zero-shot neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4465–4470, Online. Association for Computational Linguistics.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2018. Efficient parametrization of multi-domain deep neural networks. *CoRR*, abs/1803.10082.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019a. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *CoRR*, abs/1907.05791.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin. 2019b. Ccmatrix: Mining billions of high-quality parallel sentences on the WEB. *CoRR*, abs/1911.04944.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Asa Cooper Stickland, Xian Li, and Marjan Ghazvininejad. 2020. Recipes for adapting pre-trained monolingual and multilingual models to machine translation. *CoRR*, abs/2004.14911.

Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. *CoRR*, abs/1902.10461.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Pavanpankaj Vegi, Sivabhavani J, Biswajit Paul, Abhinav Mishra, Prashant Banjare, Prasanna Kumar K R, and Chitra Viswanathan. 2022. Webcrawl african: A multilingual parallel corpora for african languages. In *Proceedings of the Seventh Conference on Machine Translation: Shared Task Papers*.

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020. Balancing training for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8526–8537, Online. Association for Computational Linguistics.

Yining Wang, Jiajun Zhang, Feifei Zhai, Jingfang Xu, and Chengqing Zong. 2018. Three strategies to improve one-to-many multilingual translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2955–2960, Brussels, Belgium. Association for Computational Linguistics.

Barack Wanjawa, Lilian Wanzare, Florence Indede, Owen McOnyango, Edward Ombui, and Lawrence Muchemi. 2022. Kencorpus: A kenyan language corpus of swahili, dholuo and luhya for natural language processing tasks.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021. Share or not? learning to schedule language-specific capacity for multilingual translation. In *International Conference on Learning Representations*.

Appendix

	BLEU							
	Fine-Tune			Language-Tune			Family-Tune	
Pairs	Devtest	Test	Tico	DevTest	Test	Tico	Devtest	Test
eng-afr	40.1	39.3		40.5	39.8		40.2	39.6
eng-amh	11.5	7.5	10.5	11.7	7.6	10.3	11.1	7.3
eng-fuv	0.2	0.4	0.2	0.3	0.3	0.3	0.3	0.3
eng-hau	10.1	10.4	3.4	12.8	13.3	5.6	13.5	14.5
eng-ibo	15.1	16.8		15.8	17.3		15.9	17.3
eng-kam	2.8	2.8		2.8	2.9		3	3
eng-lug	6	6.1	11.3	5.4	5.8	11	5.4	5.6
eng-luo	7.3	7.6		7.9	8		8.1	8.1
eng-nso	22.8	23.4		22.6	23.5		22.2	23
eng-nya	13.7	13		14.2	13.3		14	13.4
eng-orm	1.3	1.6	3.3	1.3	1.4	3.3	1.4	1.5
eng-kin	12.7	13.6	13.6	12.4	13.2	13.7	12.4	12.8
eng-sna	10.2	10		11	10.6		10.6	10.6
eng-som	10.9	12	8.3	11.1	11.9	8.5	11.1	11.9
eng-ssw	7.7	7.5		7.6	7.2		7.1	6.8
eng-swh	33.2	31.6	30.8	33.7	32.7	31.3	33.6	32.6
eng-tsn	17	18		18.6	19.7		17.6	19.1
eng-tso	15.1	16.1		16.3	17.4		16	17.2
eng-umb	1	0.8		1.1	0.8		1.4	0.9
eng-xho	1.3	1		1.4	1		1.7	1.4
eng-yor	3.3	3.1		3.4	3.2		3.3	3.2
eng-zul	15.8	13.1	16.8	16.1	13.2	17.2	16.1	13.5
afr-eng	55.1	56		56.5	57		56.3	57
amh-eng	30.5	29.5	27.6	31.3	30.7	28.6	31.2	30.1
fuv-eng	6.1	6.6	12.5	6.8	6.9	13	6.1	6.7
hau-eng	28.1	29.8	30.9	28	29.6	30.9	27.3	29.1
ibo-eng	25.2	28		25.8	28.2		25.6	28.2
kam-eng	9.4	10.7		9.5	10.9		9.7	10.9
lug-eng	15.3	16.5	25.8	16.2	16.8	26.7	16.3	17.2
luo-eng	17.3	19		18	19.2		18.2	19.1
nso-eng	33.1	33.3		34.4	34.7		34.4	35.2
nya-eng	24.9	25.8		25.2	25.8		24.9	25.8
orm-eng	12.2	13.3	17.8	13.2	14.6	18.8	13.3	14.6
kin-eng	27.5	28	22.7	28.3	28.5	22.9	28.1	28.3
sna-eng	25	25.8		25.3	26.1		25.4	26.3
som-eng	23.7	26.1	14.7	24	26.4	15	24.2	26.4
ssw-eng	26.3	27.1		25.8	27.1		25.8	27.1
swh-eng	41.3	41.1	40	41.4	41.1	40.5	41.8	41

	BLEU							
	Fine-Tune			Language-Tune			Family-Tune	
Pairs	**Devtest**	**Test**	**Tico**	**DevTest**	**Test**	**Tico**	**Devtest**	**Test**
tsn-eng	23.7	25.7		23.9	25.6		23.9	26.6
tso-eng	27	27.5		28	28.1		27.6	28.3
umb-eng	7.1	7.6		7.1	7.7		7.9	8
xho-eng	34.5	31.2		35.2	31.3		34.9	31.3
yor-eng	16.1	17.1		16.7	17.6		16.6	17.6
zul-eng	35.4	33.9	40.2	35.8	34.4	40.6	36	34.6
fra-kin	9.4	10.1	10.8	9.5	10.3	11	9.4	10.1
fra-lin	6.3	6.5	6.8	7	7.2	7.5	7.4	7.5
fra-swh	22.5	21.7	20.4	23.6	22.8	20.8	23.9	23.4
fra-wol	1.8	1.8		1.8	1.8		2.1	2
kin-fra	22.5	22.7	18.4	22.7	22.7	18.7	22.9	23
lin-fra	18.1	17.9	16.4	18.6	19.1	16.9	18.4	18.8
swh-fra	31.2	30.6	26.1	31.8	31	26	31.5	30.8
wol-fra	9	9.9		9.9	10.5		9.7	10.3
xho-zul	12.4	9.9		12.9	10		12.9	9.9
zul-sna	9.5	9.3		9.5	9.9		9.6	9.9
sna-afr	16.2	16.8		16.2	17		16.3	17
afr-ssw		6.9			6.6		5.9	6.5
ssw-tsn		16.4			16.5		14.7	15.9
tsn-tso	11.9	13.4		12.6	13.5		11.3	12.9
tso-nso	16.9	17.4		17.2	17.8		16.9	17.9
nso-xho	10.3	8.7		10.2	8.5		10.5	8.7
swh-amh	8.5	6	7.6	8.4	5.9	7.4	8.3	5.8
amh-swh	20	18.5	17.2	20.2	18.5	17.6	20.1	18.6
luo-orm	0.5	0.6		0.5	0.7		0.5	0.7
som-amh	5.2	4.1	3	5.2	4.1	3	5.3	4.1
orm-som	4.4	5	4	4.8	5.4	4.2	4.7	5.4
swh-luo	5.3	5.6		6.4	6.5		6.6	6.6
amh-luo	4.4	4.9		4.9	5		4.9	4.7
luo-som	5.3	5.8		5.5	6.3		5.5	6.1
hau-ibo	11.6	13.2		11.6	13.4		11.6	13.5
ibo-yor	2.2	2.4		2.2	2.5		2.3	2.5
yor-fuv	0.1	0.2		0.2	0.3		0.1	0.3
fuv-hau	2.3	2.4	5	2.6	2.7	5.8	2.3	2.5
ibo-hau	13.8	14.7		13.6	14.7		13.6	14.9
yor-ibo	8.4	9		8.5	9.3		8.3	9.3
fuv-yor	0.3	0.4		0.4	0.4		0.6	0.6
hau-fuv	0.3	0.3	0	0.1	0.3	0.4	0.1	0.3

	BLEU							
	Fine-Tune			Language-Tune			Family-Tune	
Pairs	**Devtest**	**Test**	**Tico**	**DevTest**	**Test**	**Tico**	**Devtest**	**Test**
wol-hau	4.8	5.5		5.1	5.7		5.1	5.8
hau-wol	2.2	2.5		2.2	2.3		2.1	2.4
fuv-wol	0.7	1		0.8	0.8		0.7	0.9
wol-fuv	0.1	0.2		0.1	0.3		0.1	0.3
kin-swh	19.3	18.7	16.4	19.8	19.3	17	19.8	19.3
lug-lin	5.4	5.5	9.3	5.2	5.7	8.5	5	5.5
nya-kin	8.9	9.1		9.2	9.3		8.9	9
swh-lug	4.4	4.7	8.5	4.7	5	9.6	4.8	5.5
lin-nya	7.3	7.9		7.8	8		7.7	8.1
lin-kin	7.9	8.3	9.5	8.3	8.4	9.9	8.1	7.9
kin-lug	2.6	2.6	4.2	2	1.9	3.5	2	2
nya-swh	17.5	17		17.8	17.2		17.9	17.2
amh-zul	8.5	7.5	8.5	8.8	7.3	9	8.8	7.4
yor-swh	11.4	11.2		11.9	11.6		11.8	11.5
swh-yor	2.6	2.7		2.7	2.8		2.7	2.7
zul-amh	7.8	4.9	7.5	7.6	5	7.6	8.1	5.1
kin-hau	14.5	15.6	12.9	14.9	16.5	13.3	15	16.7
hau-kin	10.3	11	10.7	10.2	10.9	10.8	10.1	11.1
nya-som	7.3	8		7.2	8.1		7.3	8
som-nya	9.2	9.8		9.4	9.7		9.4	9.7
xho-lug	3.9	4.2		4	4.2		4.2	4.4
lug-xho	4.9	4.5		5.1	4.7		5	4.6
wol-swh	6.6	6.6		6.7	6.9		6.7	6.6
swh-wol	2.1	2.3		2.4	2.3		2.3	2.5

Table 5: BLEU scores of our multilingual models on all translation directions.

	CHRF++							
	Fine-Tune			Language-Tune			Family-Tune	
Pairs	**Devtest**	**Test**	**Tico**	**DevTest**	**Test**	**Tico**	**Devtest**	**Test**
eng-afr	65.4	65		65.8	65.4		65.5	65.2
eng-amh	35.9	32.4	31.9	36.5	32.7	31.6	35.7	32.2
eng-fuv	11.6	11.7	11	11.5	11.6	11	11.8	11.9
eng-hau	22	22.3	9.9	27.3	27.7	14.1	28.4	29.7
eng-ibo	38.2	39.9		39.5	40.9		39.6	40.7
eng-kam	19.4	19.2		19.2	19.2		19.2	19.3
eng-lug	30	30.6	32.7	29.4	30.7	32.4	28.8	29.6
eng-luo	29.3	29.7		30.8	30.9		30.9	30.8
eng-nso	47.7	47.2		47.8	47.9		46.9	47.2
eng-nya	43.8	43.4		44.4	44		44.2	43.9
eng-orm	17.6	18.3	19.3	18.1	18.5	19.3	18	18.3
eng-kin	37.7	38.7	39.5	37.8	38.2	39.4	37.6	37.6
eng-sna	40.6	40.3		41.3	40.9		41.1	40.9
eng-som	40.1	41.2	29.9	40.8	41.6	30.1	40.7	41.5
eng-ssw	38.6	39.1		38.9	39.4		38.1	38.4
eng-swh	58.7	57.9	56.2	59.3	58.7	56.6	59.3	58.4
eng-tsn	40.8	40.9		43.1	43.7		41.9	43
eng-tso	42.4	42.4		43.7	44.2		43.1	43.8
eng-umb	18.7	18.2		19.3	19		20	19.5
eng-xho	15.2	14.2		15.7	14.7		17.6	17.2
eng-yor	19.3	19.5		19.6	19.6		19.5	19.7
eng-zul	49.4	47.3	49.9	50.1	47.8	50.4	50	47.7
afr-eng	73.6	74.2		74.3	75		74.3	74.9
amh-eng	54.6	53.2	51.6	55.4	54.2	52.6	55.3	53.7
fuv-eng	22.4	22.4	28.9	23.4	23.4	29.9	22.5	22.8
hau-eng	49.7	51	51.6	50.1	51.4	51.8	49.7	50.9
ibo-eng	47.4	50		48.5	50.6		48.1	50.4
kam-eng	27.3	28.1		28.5	29		28.6	29.1
lug-eng	35.8	36	45.6	36.7	36.6	46.6	36.9	37.1
luo-eng	39	39.2		39.4	39.7		39.5	39.4
nso-eng	53.7	53.4		54.9	54.8		54.9	55.3
nya-eng	47.3	47.6		47.8	48.1		47.5	48.2
orm-eng	33.1	33.6	38.5	34.4	35.4	39.9	34.8	35.2
kin-eng	49.2	49.3	44.7	50.1	50	44.9	49.8	49.9
sna-eng	47.8	47.9		48.1	48.1		48.2	48.4
som-eng	45.5	46.4	32.1	46.2	46.8	32.4	46.3	46.9
ssw-eng	47.7	48.2		47.4	48.2		47.6	48.5
swh-eng	62.3	61.4	60.7	62.4	61.7	61.4	62.6	61.7

	CHRF++							
	Fine-Tune			Language-Tune			Family-Tune	
Pairs	**Devtest**	**Test**	**Tico**	**DevTest**	**Test**	**Tico**	**Devtest**	**Test**
tsn-eng	45.5	46.9		46.4	47.5		47	48.9
tso-eng	48.6	48.2		49.6	49		49.3	49.3
umb-eng	25.5	25.8		25.6	26.1		26.9	26.4
xho-eng	55.7	52.6		56.4	53		56.1	52.9
yor-eng	37.7	37.8		38.6	38.7		38.6	38.6
zul-eng	57.1	54.8	61.2	57.6	55.8	61.6	57.8	55.7
fra-kin	35.4	35.8	35.4	36.4	37.2	35.7	36.1	36.4
fra-lin	32	31.9	30.4	34.1	34.3	32.7	34.7	34.6
fra-swh	49	47.9	45.8	50.8	50.1	46.3	51.1	50.5
fra-wol	11.7	11.8		12.4	12.7		14.6	14.6
kin-fra	45.3	45.1	39.8	45.6	45.7	40.2	46	45.9
lin-fra	40.2	39.8	37	40.9	41	37.5	40.8	40.7
swh-fra	53.8	53.4	48.6	54.5	53.9	49	54.4	53.9
wol-fra	28	27.9		29.6	29.2		29.2	28.7
xho-zul	45.3	43.2		45.8	43.3		45.7	43.1
zul-sna	40.5	39.9		40.6	40.4		40.6	40.4
sna-afr	41.6	41.4		42	41.9		42	41.9
afr-ssw		39.3			39		37.3	38.5
ssw-tsn		40.7			40.9		39.5	40.4
tsn-tso	38.4	40		39.7	40.5		38.7	40.1
tso-nso	41.8	41.9		42	42.3		42	42.3
nso-xho	41.8	40		41.8	40.3		42	40.4
swh-amh	31.7	29.2	27.1	31.9	29.2	26.8	31.9	29.1
amh-swh	48.2	46.4	44.3	48.3	46.7	44.9	48.4	46.7
luo-orm	14.5	15.1		14.8	15.6		14.6	15.6
som-amh	24.2	22.9	14.4	24.4	23	14.5	24.3	23.2
orm-som	27.9	29	23.1	29	29.6	23.4	28.9	29.8
swh-luo	26.6	26.8		28.6	28.9		28.9	28.9
amh-luo	26.1	26		26.4	26.6		26.7	26.2
luo-som	29.8	30.2		30.3	31		29.9	30.6
hau-ibo	34.1	35.3		34.1	35.7		34.1	35.7
ibo-yor	17.4	18		17.4	18.2		17.6	18.4
yor-fuv	11.2	11.2		11.1	11.1		11.2	11.2
fuv-hau	16.9	17.1	19.7	17.2	17.6	21.2	16.3	16.8
ibo-hau	38.5	39.7		38.7	39.9		38.5	40.1
yor-ibo	29.6	30		29.8	30.5		29.6	30.4
fuv-yor	6.4	6.5		7	7		8	8.1
hau-fuv	11.4	11.5	10.7	11.1	11.2	10.5	11.4	11.5

	CHRF++							
	Fine-Tune			**Language-Tune**			**Family-Tune**	
Pairs	**Devtest**	**Test**	**Tico**	**DevTest**	**Test**	**Tico**	**Devtest**	**Test**
wol-hau	23.4	23.6		24.2	24.2		22.9	23.3
hau-wol	13.4	14.1		13.2	13.6		13.6	14.5
fuv-wol	8.5	9		9.2	9.1		9.6	9.5
wol-fuv	11.5	11.7		11.4	11.5		11.7	11.8
kin-swh	45.9	45.8	42.5	46.4	46.6	43.2	46.4	46.3
lug-lin	28.5	28.8	32.6	29.5	29.8	33.2	29.3	29.3
nya-kin	34.6	34.3		35.2	35		34.4	34.4
swh-lug	27.5	28.2	30.3	29	29.4	31.7	29.8	30.3
lin-nya	34.2	34.7		34.9	35.2		35.1	35.5
lin-kin	32.5	32.7	33.2	33.3	33.2	33.6	32.9	32.7
kin-lug	21.6	21.6	21.5	19.3	19.3	20	19.3	19.2
nya-swh	44.6	44.3		44.9	44.7		44.9	44.7
amh-zul	41.4	39.9	39	42.1	40.4	40	41.9	40.3
yor-swh	36.8	36.4		37.5	37.2		37.3	36.8
swh-yor	18.4	18.5		18.3	18.7		18.5	18.7
zul-amh	30.1	26.2	27.2	30.2	26.7	27.3	30.4	26.7
kin-hau	38.8	40	36.4	39.7	41.4	37.1	39.6	41.5
hau-kin	36.2	36.7	35.5	36.3	36.7	35.7	36.2	37
nya-som	34.6	35.9		35	36.2		35	36.1
som-nya	37.5	37.9		37.6	38		37.6	38.1
xho-lug	26.2	26.8		26.5	27		26.9	27.4
lug-xho	31.2	29.9		32	30.8		31.8	30.8
wol-swh	28.3	27.2		28.8	28.3		28.5	27.4
swh-wol	13.1	13.4		14.2	13.7		14.5	14.6

Table 6: CHRF++ scores of our multilingual models on all translation directions.

	spBLEU					
	Fine-Tune		Language-Tune		Family-Tune	
Pairs	Devtest	Test	Devtest	Test	Devtest	Test
eng-afr	45.6	44.7	46.1	45.2	45.7	44.9
eng-amh	26.1	21.8	26.7	22.1	25.9	21.5
eng-fuv	0.4	0.6	0.4	0.5	0.4	0.5
eng-hau	3	3.1	4.3	4.5	4.7	5.2
eng-ibo	17.6	18.9	18.6	19.6	18.7	19.6
eng-kam	3.7	3.8	3.8	4	3.9	4
eng-lug	7.8	7.9	6.8	7.5	6.6	7
eng-luo	9.5	9.8	10.2	10.4	10.3	10.4
eng-nso	24.1	24.4	24.3	24.8	23.9	24.4
eng-nya	17.3	16.9	18	17.3	17.8	17.2
eng-orm	2.3	2.6	2.4	2.4	2.3	2.4
eng-kin	16	16.6	15.8	16.4	15.8	16.2
eng-sna	16.2	15.9	17.3	16.8	16.9	16.7
eng-som	16	17.2	16.4	17.5	16.3	17.3
eng-ssw	14.8	15.3	14.6	15.1	14.2	14.4
eng-swh	37.2	35.4	38	36.5	37.8	36.2
eng-tsn	18.5	19	20.1	20.7	19.1	20.1
eng-tso	18	18.9	19.5	20.5	19.1	20.1
eng-umb	1.9	1.9	2.1	2.1	2.3	2.2
eng-xho	3.3	2.5	3.4	2.7	4.1	3.5
eng-yor	4.6	4.6	5.2	4.8	4.9	4.9
eng-zul	26.2	23.3	27.1	23.9	27.1	24
afr-eng	58.2	58.8	59.5	60.1	59.4	60
amh-eng	33.1	31.2	33.9	32.3	33.7	31.6
fuv-eng	7.8	8.1	8.6	8.5	8	8.5
hau-eng	31.1	32.4	31.1	32.4	30.5	31.8
ibo-eng	28.1	30.7	28.8	30.8	28.5	30.8
kam-eng	11.9	12.9	12.3	13.2	12.2	13.2
lug-eng	17.4	18.4	18.4	18.7	18.3	19
luo-eng	20.3	20.9	20.7	21.2	20.7	21
nso-eng	35.3	35	36.6	36.7	36.6	37.1
nya-eng	28.1	28.6	28.6	28.8	28.2	29
orm-eng	13.2	14	14.4	15.3	14.6	15.2
kin-eng	29.5	29.7	30.3	30.2	30.1	30
sna-eng	28.7	29	29	29.3	29.2	29.4
som-eng	25.8	27.5	26.1	27.8	26.4	27.8
ssw-eng	28.6	29	28.2	29	28.3	29.2
swh-eng	43.3	42.5	43.3	42.7	43.6	42.8

	spBLEU					
	Fine-Tune		Language-Tune		Family-Tune	
Pairs	**Devtest**	**Test**	**Devtest**	**Test**	**Devtest**	**Test**
tsn-eng	26.3	27.8	27	28.2	27.4	29.3
tso-eng	29.7	29.4	30.8	30.2	30.4	30.3
umb-eng	8.9	9.5	9	9.6	9.8	9.9
xho-eng	37.3	33.6	38.1	33.9	37.8	33.7
yor-eng	18.2	19	19	19.7	18.9	19.6
zul-eng	38.3	35.9	38.9	36.8	39	36.9
fra-kin	12.9	13.6	13.4	14.4	13.3	13.9
fra-lin	8.8	9	9.6	10.1	9.8	10.1
fra-swh	26.6	25.3	28.1	27.1	28.5	27.6
fra-wol	2.1	2.3	2.6	2.6	3.2	3
kin-fra	26.3	25.9	26.7	26	26.9	26.4
lin-fra	22.5	21.9	23	23	23	22.8
swh-fra	35.7	34.8	36.2	35	36	34.9
wol-fra	12.7	12.6	13.7	13.2	13.3	13
xho-zul	22.6	19.9	23.2	20.1	23.2	20
zul-sna	16.5	16.1	16.6	16.6	16.7	16.7
sna-afr	19.7	19.5	20.1	20	20.1	20.1
afr-ssw		15.2		14.7	13.1	14.3
ssw-tsn		17.6		17.7	16.2	17.1
tsn-tso	14.8	15.9	15.8	16.1	14.3	15.4
tso-nso	18.3	18.6	18.9	18.9	18.7	19.2
nso-xho	17.5	15.9	17.3	16	17.5	15.9
swh-amh	21.6	18.5	21.8	18.5	21.9	18.3
amh-swh	24.1	21.6	24.4	21.8	24.2	21.9
luo-orm	1.2	1.2	1.2	1.3	1.1	1.3
som-amh	14.7	13.2	14.9	13.2	14.9	13.4
orm-som	6.7	7.4	7.3	7.8	7.3	7.9
swh-luo	7.4	7.5	8.4	8.7	8.8	8.8
amh-luo	6	6.3	6.5	6.4	6.6	6.1
luo-som	8.2	8.5	8.5	9.1	8.4	8.9
hau-ibo	14.2	15.5	14.3	15.7	14.3	15.6
ibo-yor	3.6	3.8	3.8	3.9	3.8	4
yor-fuv	0.2	0.3	0.3	0.4	0.3	0.4
fuv-hau	2.8	2.9	2.9	3.1	2.8	3.1
ibo-hau	16.3	16.8	15.9	16.7	15.8	17
yor-ibo	10.9	11.3	11.1	11.5	11	11.5
fuv-yor	0.6	0.6	0.7	0.7	1	1
hau-fuv	0.3	0.4	0.2	0.4	0.2	0.4

	spBLEU					
	Fine-Tune		**Language-Tune**		**Family-Tune**	
Pairs	**Devtest**	**Test**	**Devtest**	**Test**	**Devtest**	**Test**
wol-hau	6.1	7	6.5	6.9	6.4	7.2
hau-wol	3.2	3.5	3.3	3.5	3.2	3.7
fuv-wol	1	1.3	1.1	1.1	1.1	1.3
wol-fuv	0.1	0.4	0.2	0.4	0.2	0.4
kin-swh	22.8	21.8	23.3	22.6	23.3	22.4
lug-lin	6.7	6.8	6.9	7.2	6.7	7
nya-kin	12.1	11.8	12.6	12.4	12	11.9
swh-lug	5.8	6.2	6.2	6.5	6.7	7.1
lin-nya	9.8	10.4	10.4	10.5	10.4	10.6
lin-kin	10.3	10.7	10.9	11	10.8	10.5
kin-lug	4.4	4.3	3.4	3.2	3.4	3.2
nya-swh	21.4	20.5	21.7	20.7	21.7	20.7
amh-zul	17.6	15.4	18.1	15.8	18.1	15.7
yor-swh	14.1	13.4	14.6	14	14.6	13.7
swh-yor	3.7	3.7	3.8	3.9	3.9	3.7
zul-amh	20.4	16.2	20.6	16.4	20.7	16.6
kin-hau	16.9	17.7	17.5	18.7	17.4	18.9
hau-kin	13.5	14.2	13.6	14.2	13.4	14.3
nya-som	11.5	12.5	11.6	12.6	11.7	12.4
som-nya	12.4	12.7	12.6	12.8	12.4	12.8
xho-lug	5.3	5.5	5.3	5.5	5.4	5.7
lug-xho	10	9.1	10.4	9.5	10.2	9.6
wol-swh	8.6	8.2	8.9	8.7	8.7	8.3
swh-wol	3	3.1	3.6	3.4	3.6	3.5

Table 7: spBLEU scores of our multilingual models on all translation directions.

Samsung Research Philippines - Datasaur AI's Submission for the WMT22 Large Scale Multilingual Translation Task

Jan Christian Blaise Cruz
Samsung Research Philippines
Manila, Philippines
jcb.cruz@samsung.com

Lintang Sutawika[*]
Datasaur AI
San Francisco, California, USA
lintang@datasaur.ai

Abstract

This paper describes the submission of the joint Samsung Research Philippines - Datasaur AI team for the WMT22 Large Scale Multilingual African Translation shared task. We approach the contest as a way to explore task composition as a solution for low-resource multilingual translation, using adapter fusion to combine multiple task adapters that learn subsets of the total translation pairs. Our final model shows performance improvements in 32 out of the 44 translation directions that we participate in when compared to a single model system trained on multiple directions at once.

1 Introduction

In this paper, we describe two systems that we submit to the WMT22 Large Scale Multilingual African Translation shared task: a baseline finetuned MT5 (Xue et al., 2020) model trained on multiple directions at once (referred to as `SRPH-DAI-Baseline`), and an MT5 model successively finetuned with task composition using multiple pair-specific adapters (referred to as `SRPH-DAI-Fusion`).

We first outline the preprocessing steps and filtering heuristics used to clean the contest dataset, then we show the training setup and experimental design used for constructing our submitted systems. We then report our results on the hidden test set via BLEU, spBLEU, and CHRF2++ automatic evaluation metrics.

2 Preprocessing

In this section, we detail the preprocessing steps used to filter the contest dataset to ensure that data quality is as high as possible.

Given that the contest dataset contains sentence pairs that were artificially aligned from crawled data, we use a number of filters to reduce the possibility of mismatched pairs in the final training dataset:

- We filter out pairs where one or both sentences have too few ($<= 3$) or too many ($>= 150$) tokens post-sentencepiece tokenization.
- We remove pairs if one or both sentences have too many repeated ($>= 5$) punctuations or symbols of the same type (e.g. "/////"), or contiguous punctuations/symbols of considerable ($>= 3$) length (e.g. "`word` $&**$").
- We also remove sentence pairs where one sentence has punctuation that is missing from the other (e.g. "`word`!!" → "`word`?").
- If a pair has a sentence where a large percentage of the total characters (total $>= 70\%$) are numbers or punctuations (e.g. "`word` ??! +22 8456 8967"), the pair is dropped.
- An average word length filter is also used to remove pairs where one sentence has words that are disproportionately longer than the words in the corresponding sentence. We get a ratio r by taking the sum of the lengths of each token in a sentence, then dividing it by the number of tokens. We only keep sentence pairs where both sentences have a ratio r within $3 <= r <= 15$.
- HTML and URL-containing sentence pairs are also removed as this contributes to unnecessary noise during training.
- Lastly, we also check for known word matches within each sentence pair. For instance, if we detect a number (e.g. "1" or "one"), we also check the corresponding sentence for the same number. Sentence pairs that have mismatched

[*] Work done while at Konvergen AI.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1034 - 1038
December 7-8, 2022 ©2022 Association for Computational Linguistics

Pair	Samples
afr ↔ eng	2,526,513
amh ↔ eng	315,870
fuv ↔ eng	953,002
hau ↔ eng	1,841,974
ibo ↔ eng	136,534
kam ↔ eng	1,143,082
kin ↔ eng	7,143,167
lug ↔ eng	2,058,590
luo ↔ eng	1,713,159
nso ↔ eng	1,600,977
nya ↔ eng	1,289,859
orm ↔ eng	1,786,712
sna ↔ eng	5,917,741
som ↔ eng	413,647
ssw ↔ eng	77,807
swh ↔ eng	18,243,580
tsn ↔ eng	3,034,232
tso ↔ eng	383,586
umb ↔ eng	190,170
xho ↔ eng	5,481,855
yor ↔ eng	923,055
zul ↔ eng	2,645,396

Table 1: Final dataset statistics after running the sentence pair filters.

(e.g. source sentence has “1” but target sentence has “11”) words are dropped as these are likely from misaligned data.

After applying the filters for the entire dataset, we perform one deduplication step to ensure that no duplicate entries have been added. No further preprocessing is done on the data itself to preserve as much information within the sentences as possible.

When formatting the data for translation training, we insert a target language token at the beginning of the sentence. For example, a sentence to be translated from English to Afrikaans would look like:

<afr> This is an example sentence.

We only participate in a subset of the shared task’s translation pairs (44 total directions), opting to train only on English → African and African → English pairs due to resource constraints.

3 Experiment Design

In this section, we describe the construction of our two submitted systems: `SRPH-DAI-Baseline` and `SRPH-DAI-Fusion`.

3.1 Common Settings

Both systems use MT5-Small, a Transformer-based (Vaswani et al., 2017) model, as an initialization point. We opted to use the small variant (∼300 million parameters) as opposed to the bigger base (∼580 million) and large (∼1.2 billion) variants due to resource constraints in our setup. We expect the performance of our models to further improve as we scale to larger variants of pretrained models.

As a remedy to constrained resources as well as a way to improve stability during training for low-resource data, we decided to use adapters (Houlsby et al., 2019; Pfeiffer et al., 2020b) instead of fully finetuning all the model parameters.

Before proceeding to training for translation, we first train a language adapter (Pfeiffer et al., 2020b) on English + African languages in order to better condition the MT5 model for the languages it will encounter later. We mimic MT5’s pretraining and use span corruption on the provided monolingual training data for the shared task (which is likewise filtered like our parallel data).

We freeze the pretrained weights and train the adapter for a total of 150K steps using the Adafactor (Shazeer and Stern, 2018) optimizer, utilizing a learning rate schedule that warms up for the first 10K steps to a maximum of $1e-4$, then linearly decaying after. We use a maximum sequence length of 512 for language adapter training, using gradient accumulation to train with a total batch size of 128 sequences per training step. The output language adapter is used in both of our submission systems, and is stacked below the translation task adapter(s).

3.2 Baseline Model

We construct our baseline model `SRPH-DAI-Baseline` by stacking a blank task adapter on top of our language adapter and training it on all 44 translation directions at once. In this setup, the language adapter is frozen. This model is trained for a total of 300K steps on the combined filtered dataset using the Adafactor optimizer. We use a learning rate of $5e-5$ and a weight decay of $1e-8$, warming up for the first 10K steps, then linearly decaying after.

Unlike other systems, we do not perform any other techniques such as backtranslation (Edunov et al., 2018), noisy channel reranking (Yee et al., 2019), or clever pair sampling (Fan et al., 2021) in order to further boost performance. This mimics an "ablation" setup where only the direct finetuning method is used in order to accurately observe the effect of using task composition later on. Since no further modifications are made on the model beyond the training method, any improvements on performance made by task composition can be attributed to task composition and not anything else.

3.3 Exploring Task Composition

In the conventional multidirectional setup like in our baseline, the model learns generic cross-lingual information at the same time that it learns task-specific information. Learning cross-lingual information is useful in cases where a number of the languages in the model are similar or come from the same family (Saleh et al., 2021; Siddhant et al., 2022). However, in cases where a number of the languages are dissimilar or come from different families, we hypothesize that it may be useful to learn cross-lingual information *separately* from task-specific mappings. This ensures that the model learns each translation direction in a non-destructive manner with respect to other language pairs.

In cases where certain language pairs are underrepresented in the training set, learning each direction separately also removes the need for specialized data sampling methods to ensure that the model sees each pair enough times. In addition, using adapters for low-resource pairs also helps prevent overfitting the small dataset (Mao et al., 2021).

Motivated by this, instead of finetuning a task adapter for multiple translation directions, we instead opt to train *multiple* translation task adapters to learn task-specific information, then *composing* the multidirectional setup afterwards via Adapter Fusion (Pfeiffer et al., 2020a) to mix cross-lingual and cross-task information. This is how we construct our SRPH-DAI-Fusion model.

For this setup, we follow the same training routine as in the baseline, except we only train on one language pair at a time. We train an adapter to produce translations for *two* directions: English $\rightarrow X$ and $X \rightarrow$ English. Training in more than one direction ensures that the task adapters learn to properly embed the target language token at the beginning of every sentence. This results in a total of 22 task adapters for each of the 22 English $\rightarrow X$ pairs.

Finally, we add an Adapter Fusion setup for all 22 single-pair task adapters, freeze the adapters,then further finetune the model to learn cross-task and cross-lingual information. We finetune for 100K steps with a learning rate of $2e-5$ using the Adafactor optimizer. Like in previous setups, we also use a warmup of 10K steps with a linear decay afterwards.

4 Results

We outline the performance of our two models on the hidden test set on Table 2.

Overall, SRPH-DAI-Fusion outperforms SRPH-DAI-Base on average across all three metrics, with an improvement of 0.09, 0.19, and 1.33 on average BLEU, spBLEU, and CHRF2++, respectively. Both models perform relatively better on the African to English translation directions compared to the English to African ones. We hypothesize that this is likely due to English being a pivot language, and thus cross-lingual and cross-task information learned while training each pair contributed to better performance when translating into English.

When comparing the two models, we note an "improvement" in the performance if at least two of the three metrics had an increase in score. We observe that 32 out of the 44 translation directions had an improvement once task composition was used for finetuning, most of which are very low-resource pairs. Best gains are observed in the English to African translation directions, with some pairs such as Eng $\rightarrow$ Orm improving from an initial 0 score from the baseline model.

We observe that SRPH-DAI-Base outperforms the task composition model in cases where there is a relative abundance of training data. For pairs that have sub-million examples, SRPH-DAI-Fusion performs much better, likely due to the model being able to learn more specialized information about these translation directions separate from the other directions.

Interestingly, we observe that for language pairs with a relative abundance of data, the drop in performance when using task composition is substantial. For example, Afr $\rightarrow$ Eng suffers a 2.2, 2, and 4.2 points drop in BLEU, spBLEU, and CHRF2++, respectively. We hypothesize that this is because SRPH-DAI-Fusion has more intact task-specific knowledge related to low-resource pairs that may not be useful to the higher-resourced pairs. Since task adapters are frozen during fusion layer training, the model has an added burden in learning how to adapt knowledge that may not be useful when translating higher-resourced translation directions.

5 Conclusion

In this paper, we described our submissions for the WMT22 Large Scale Multilingual African Translation shared task. We approached the contest as a way to explore task composition as a solution for multilingual translation, especially among low-resource languages. In our experiments, we show that using task composition – training task adapters to learn pair-specific knowledge, then using a fusion layer to learn cross-task information – improves performance for less-represented language pairs in a multilingual translation dataset. While the model's results for a number of translation directions are far from state-of-the-art, the results show the methodology's promise for further exploration.

For future work, we would like to conduct experiments for larger models than is constrained by our resources. We expect that using Base and Large variants of MT5 would further improve performance for all language pairs. In addition, it would be beneficial to test the methodology while adding in common "best practices" in translation such as using backtranslated data and better data sampling. Lastly, we would like to explore setups where the pair-specific task adapters are *transformable* to some extent instead of being fully frozen as a remedy to the problem of higher-resourced pairs performing worse in the task composition setup.

References

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(107):1–48.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Wen-tau Yih, and Madian Khabsa. 2021. Unipelt: A unified framework for parameter-efficient language model tuning. *arXiv preprint arXiv:2110.07577*.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2020a. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *arXiv preprint arXiv:2005.00052*.

Fahimeh Saleh, Wray Buntine, Gholamreza Haffari, and Lan Du. 2021. Multilingual neural machine translation: Can linguistic hierarchies help? *arXiv preprint arXiv:2110.07816*.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Aditya Siddhant, Ankur Bapna, Orhan Firat, Yuan Cao, Mia Xu Chen, Isaac Caswell, and Xavier Garcia. 2022. Towards the next 1000 languages in multilingual machine translation: Exploring the synergy between supervised and self-supervised learning. *arXiv preprint arXiv:2201.03110*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Kyra Yee, Nathan Ng, Yann N Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. *arXiv preprint arXiv:1908.05731*.

	`SRPH-DAI-Base`			`SRPH-DAI-Fusion`			
Pair	**BLEU**	**spBLEU**	**CHRF2++**	**BLEU**	**spBLEU**	**CHRF2++**	**Improved?**
afr → eng	8.3	9.1	26.2	6.1	7.1	22	-
amh → eng	0.7	0.8	11.2	0.9	1	11.3	✓
fuv → eng	1.3	1.8	10.6	1.4	2	11.1	✓
hau → eng	2.7	3.7	14.8	2.5	3.6	14.6	-
ibo → eng	1.9	2.6	12.3	2.1	3	12.7	✓
kam → eng	2.1	2.8	12.1	2.1	2.9	12.6	✓
kin → eng	2.3	3.1	14.2	2.7	3.4	15	✓
lug → eng	1.8	2.4	11.9	2	2.6	13	✓
luo → eng	1.8	2.2	11	1.8	2.4	11.7	✓
nso → eng	2.8	3.6	14.3	3.1	4.2	15.9	✓
nya → eng	3	3.9	15.4	3.1	4.2	15.9	✓
orm → eng	0.5	0.7	8.4	0.6	0.9	9.2	✓
sna → eng	3	3.7	15.1	3	3.7	15.5	-
som → eng	2	2.5	12.4	2.3	3	14	✓
ssw → eng	2.6	3.3	13.8	2.6	3.3	14	✓
swh → eng	4.1	4.4	18.1	3.9	4.6	17.8	-
tsn → eng	2.3	2.9	13.3	2.6	3.3	14.1	✓
tso → eng	2.1	2.8	12.3	2.4	3	13.1	✓
umb → eng	1	1.5	10.7	0.9	1.5	11	-
xho → eng	3.3	4.1	16.4	3.2	4	16.3	-
yor → eng	1.5	2.1	10.9	1.8	2.5	12.2	✓
zul → eng	2.9	3.5	15.4	2.9	3.5	15.4	-
eng → afr	4.1	4.3	20.6	2.6	3	17.9	-
eng → amh	0.1	0	2.6	0.3	0.2	0.5	✓
eng → fuv	0.1	0.1	4	0.9	1.2	10	✓
eng → hau	0.3	0.5	8.3	0.5	1	19.4	✓
eng → ibo	0.2	0.1	4.4	0.6	0.8	8.7	✓
eng → kam	0.1	0.1	2.5	0.6	0.8	7.9	✓
eng → kin	0.3	0.4	5.4	0.4	0.4	8.1	✓
eng → lug	0.2	0.3	3.8	1.1	0.9	8.7	✓
eng → luo	0.5	0.6	5.7	1	1.4	10.1	✓
eng → nso	0.3	0.4	5.5	0.4	0.9	8.3	✓
eng → nya	1.1	0.8	10.6	1.4	1.4	11.5	✓
eng → orm	0	0	2.2	0.1	0.1	4.7	✓
eng → sna	1.1	0.8	11.8	1	0.8	9.3	-
eng → som	0.3	0.1	6.8	0.4	0.5	7.4	✓
eng → ssw	0.7	0.8	9	1.1	0.8	9.2	✓
eng → swh	1.3	1.4	14.9	1.1	1.6	13.2	-
eng → tsn	0.3	0.3	5.6	0.3	0.7	7.7	✓
eng → tso	0.3	0.4	3.8	0.7	1.1	8.9	✓
eng → umb	0.3	0.2	3.1	0.6	0.7	8.3	✓
eng → xho	0.6	0.9	11.1	0.6	0.6	11	-
eng → yor	0.1	0	4.1	0.3	0.3	6.3	✓
eng → zul	0.5	0.8	10.6	0.6	0.5	10.1	-
Average	1.52	1.84	10.39	1.61	2.03	11.72	

Table 2: Results of both `SRPH-DAI-Base` and `SRPH-DAI-Fusion` on the hidden test set. We consider task composition as an improvement if it resulted in an increase in performance in *at least two* of the three automatic metrics.

University of Cape Town's WMT22 System: Multilingual Machine Translation for Southern African Languages

Khalid N. Elmadani Francois Meyer Jan Buys
Department of Computer Science
University of Cape Town
{ahmkha009,myrfra008}@myuct.ac.za, jbuys@cs.uct.ac.za

Abstract

The paper describes the University of Cape Town's submission to the constrained track of the WMT22 Shared Task: Large-Scale Machine Translation Evaluation for African Languages. Our system is a single multilingual translation model that translates between English and 8 South / South East African Languages, as well as between specific pairs of the African languages. We used several techniques suited for low-resource machine translation (MT), including overlap BPE, back-translation, synthetic training data generation, and adding more translation directions during training. Our results show the value of these techniques, especially for directions where very little or no bilingual training data is available.[1]

1 Introduction

Southern African languages are underrepresented in NLP research, in part because most of them are low-resource languages: It is not always possible to find high-quality datasets that are large enough to train effective deep learning models (Kreutzer et al., 2021). The WMT22 Shared Task on Large-Scale Machine Translation Evaluation for African Languages (Adelani et al., 2022) presented an opportunity to apply one of the most promising recent developments in NLP — multilingual neural machine translation — to Southern African languages. For many languages, the parallel corpora released for the shared task are the largest publicly available datasets yet. For some translation directions (e.g. between Southern African languages), no parallel corpora were previously available.

In this paper we present our submission to the shared task. Our system is a Transformer-based encoder-decoder (Vaswani et al., 2017) that translates between English and 8 South / South East African languages (Afrikaans, Northern Sotho, Shona, Swati, Tswana, Xhosa, Xitsonga, Zulu) and in 8 additional directions (Xhosa to Zulu, Zulu to Shona, Shona to Afrikaans, Afrikaans to Swati, Swati to Tswana, Tswana to Xitsonga, Xitsonga to Northern Sotho, Northern Sotho to Xhosa). We trained a single model with shared encoder and decoder parameters and a shared subword vocabulary.

We applied several methods aimed at improving translation performance in a low-resource setting. We experimented with BPE (Sennrich et al., 2016b) and overlap BPE (Patil et al., 2022), the latter of which increases the representation of low-resource language tokens in the shared subword vocabulary. We used initial multilingual and bilingual models to generate back-translated sentences (Sennrich et al., 2016a) for subsequent training.

First, we trained a model to translate between English and the 8 Southern African languages. Then we added the 8 additional translation directions and continued training. For some of these additional directions no parallel corpora were available, so we generated synthetic training data with our existing model. By downsampling some of the parallel corpora to ensure a balanced dataset, we were able to train our model effectively in the new directions, while retaining performance in the old directions.

We describe the development of our model and report translation performance at each training stage. Our final results compare favourably to existing works with overlapping translation directions. While there is considerable disparity in performance across languages, our model nonetheless achieves results that indicate some degree of effective MT across all directions (most BLEU scores are above 10 and most chrF++ scores are above 40). We also discuss our findings regarding techniques for low-resource MT. We found overlap BPE and back-translation to improve performance for most translation directions. Furthermore, our results confirm the value of multilingual models, which proves critical for the lowest-resource languages.

[1] Our model is available at https://github.com/Khalid-Nabigh/UCT-s-WMT22-shared-task.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1039 - 1048
December 7-8, 2022 ©2022 Association for Computational Linguistics

2 Background

2.1 Multilingual Neural Machine Translation (MNMT)

Multilingual models help low-resource languages (LRLs) by leveraging the massive amount of training data available in high-resource languages (HRLs) (Aharoni et al., 2019; Zhang et al., 2020). In the context of Neural Machine Translation, a multilingual model can translate between more than two languages. Current research in MNMT can be divided into two main areas: training language-specific parameters (Kim et al., 2019; Philip et al., 2020) and training a single massive model that shares all parameters among all languages (Fan et al., 2020; NLLB Team et al., 2022). Our work lies in the second category, as we are building a single multilingual translation system by exploring back-translation and different vocabulary generation approaches.

2.2 Back-Translation

Given parallel sentences in two languages A and B (A_b, B_a), with goal of training a model that translates sentences from A to B ($A \rightarrow B$). Back-translation works as follows: First, one trains a ($B \rightarrow A$) model using the available (A_b, B_a) data. Then the B_a sentences are passed to the model to regenerate A_b. This model's output (A'_b) is then considered as additional synthetic parallel data (A'_b, B_a). The final step of back-translation is training an ($A \rightarrow B$) translation model using (A'_b, B_a) as parallel data. The motivation behind back-translation is that the noise added to the A'_b sentences from regeneration increases the model's robustness (Edunov et al., 2018). The same approach can be extended to multilingual models (Liao et al., 2021).

2.3 Overlap-based BPE (OBPE)

Byte Pair Encoding (BPE) is a vocabulary creation method that relies on n-gram frequency (Sennrich et al., 2016b). The starting point is a character-based vocabulary. At each step, the BPE algorithm identifies the two adjacent tokens with the highest frequency, joins them together as a single token, and adds the new token to the vocabulary. The dataset is then restructured based on the expanded vocabulary. In the case of multilingual training, a single BPE vocabulary can handle all languages by running the BPE algorithm on the union of the data from all the languages. However, when constructing a multilingual vocabulary, BPE will prefer frequent word types, most of which are from HRLs, leaving a smaller proportion of the vocabulary for words from LRLs.

Language Pairs	WMT22_african
eng-sna	8.7M
eng-xho	8.6M
eng-tsn	5.9M
eng-zul	3.8M
eng-nso	3M
eng-afr	1.6M
eng-tso	630K
eng-ssw	165K
xho-zul	1M
zul-sna	1.1M
sna-afr	1.6M*
afr-ssw	165K*
ssw-tsn	85K
tsn-tso	285K
tso-nso	212K
nso-xho	200K

Table 1: Number of available parallel sentences for all language pairs. * indicates that no data is available for these pairs and the number represents the amount of synthetic data we generated.

Language Family	L_{HRL}	L_{LRL}
Germanic	English(eng)	Afrikaans(afr)
Nguni	Xhosa(xho)	Zulu(zul), Swati(ssw)
Sotho-Tswana	Tswana(tsn)	Sepedi(nso)
Bantu	Shona(sna)	Xitsonga(tso)

Table 2: The languages included in our translation system, grouped by language family and whether they are used as L_{HRL} or L_{LRL} for the OBPE algorithm.

Overlap-based BPE (OBPE) is a modification to the BPE vocabulary creation algorithm which enhances overlap across related languages (Patil et al., 2022). OBPE takes into account the frequency of tokens as well as their existence among different languages. Given a list of HRLs (L_{HRL}) and LRLs (L_{LRL}), OBPE tries to balance cross-lingual sharing (tokens shared between HRLs and LRLs) and individual languages' representation. The optimal OBPE vocabulary for a set of languages from different families is produced by considering the highest resource language from each family as L_{HRL} and the rest of the languages as L_{LRL}.

3 Datasets

The WMT22 dataset is released along with the shared task. It contains bitext for 248 pairs of African languages, referred to as `WMT22_african`.[2] We use `WMT22_african` for both training and validation; the first 3 000 sentences from each language pair is reserved for validation and the rest for training. Table 1 shows available number of sentences for each language pair. No data was provided for Shona-Afrikaans and Afrikaans-Swati, so we generated synthetic data for these translation directions (see section 4.2.1). For testing, we used the Flores dev set, which contains 997 parallel sentences for each language pair. Additionally, we report the results of the final translation system as evaluated by the shared task organizers on a hidden test set.

3.1 OBPE

We trained BPE and OBPE tokenizers using the $eng \leftrightarrow$ LRL data only (the first 8 rows of table 1). The vocabulary size for both BPE and OBPE is set to 40K. For OBPE, the L_{HRL} contains the highest-resource language from each language family (eng, xho, tsn, sna), while L_{LRL} includes the rest of the languages (see table 2). We used Patil et al.'s (2022) implementation for both BPE and OBPE. This implementation is based on the Hugging Face Tokenizers library.[3]

4 Methodology

In this work, we only focus on South and South East African languages, their translation to/from English, and eight translation directions between these languages. We divided the training of the translation system into two stages. In the first stage, we trained a multilingual model for translating from all LRLs to English and vice versa. To incorporate the translation directions between LRLs into the system, we did further training on the translation model from stage 1. We divided the training process into stages instead of training the model in one session due to computational resource constraints. Both stages are explained in more detail below.

All models were trained with the `Fairseq` toolkit (Ott et al., 2019). We used the `transformer-base` architecture (Vaswani et al., 2017) for training all bilingual models. We base the multilingual models on the `BART` architecture (Liu et al., 2020), using Tang et al.'s (2021) implementation and hyperparameters, including adding a token to indicate the source language before the input sentence and a token for the target language before the output sentence.

Data	Δ
sna-eng	0.1
xho-eng	0.2
tsn-eng	−0.2
zul-eng	−0.7
nso-eng	0.3
afr-eng	0.0
tso-eng	0.0
ssw-eng	0.3
eng-sna	0.1
eng-xho	−0.2
eng-tsn	0.2
eng-zul	0.1
eng-nso	−0.2
eng-afr	0.0
eng-tso	−0.2
eng-ssw	0.0

Table 3: BLEU score differences between the OBPE multilingual model (13th epoch) and the BPE multilingual model (10th epoch) on Flores dev set. We stopped training the BPE model at this point as the OBPE model is computationally more efficient. The translation directions are sorted based on the available amount data.

4.1 Stage 1: Translation Between LRLs and English

We used BPE and OBPE vocabularies to train two multilingual models for all directions between English and LRLs. Bilingual models were trained for each translation direction using a single vocabulary for each model. Finally, we performed back-translation for all directions using the model with the highest BLEU score in each case.

4.1.1 Multilingual Training

Multilingual models generally have more parameters and require more training time and computational resources than bilingual models. Computational constraints prevented us from fully training two multilingual models and then doing back-translation from them. Subsequently we used BPE and OBPE vocabularies to train two multilingual models till the 10th and 13th epochs, respectively. At this point, we found that the difference in trans-

[2] https://huggingface.co/datasets/allenai/wmt22_african

[3] https://huggingface.co/docs/tokenizers

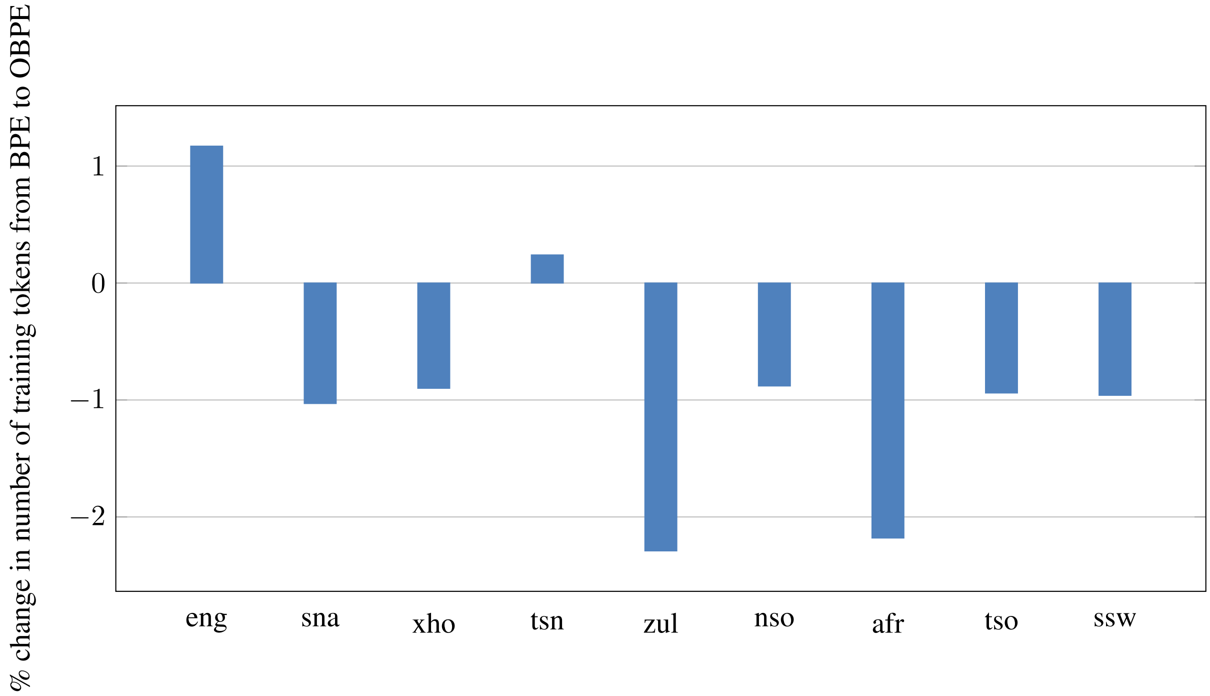


Figure 1: The change in the number of tokens in the training set per language when using OBPE instead of BPE. Less training tokens correspond to better a representation of a language in the shared subword vocabulary, so negative percentage changes reflect an improvement in low-resource language representation.

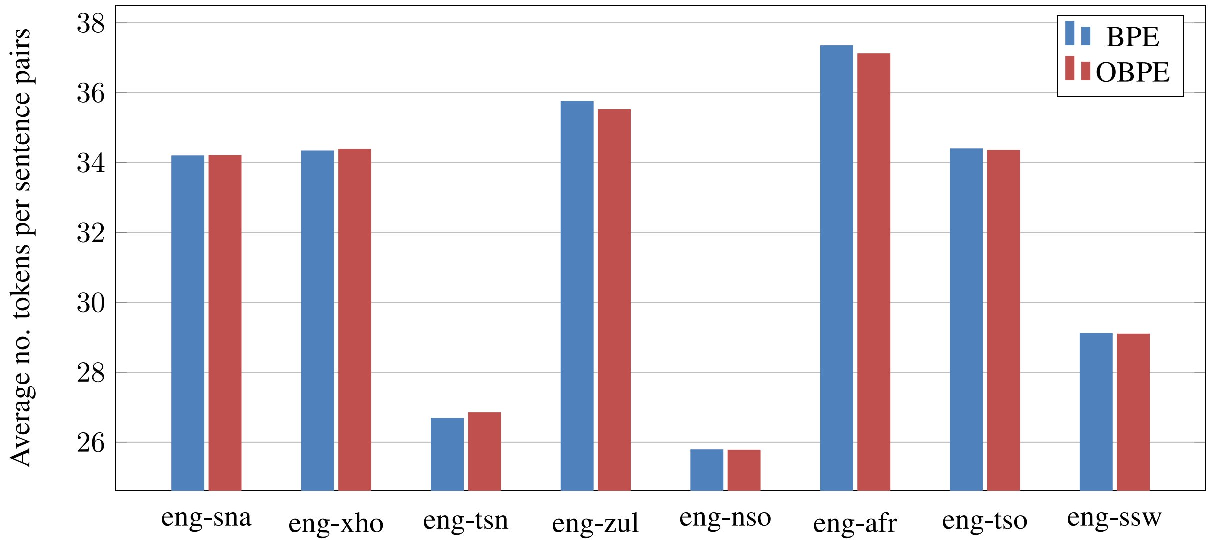


Figure 2: The average number of tokens per sentence pair for all language pairs with English, comparing BPE and OBPE vocabularies. More tokens lead to slower training.

lation quality between the two models is negligible (see table 3). However, the OBPE model is slightly faster in training and represent LRLs better. A language l is represented better in vocabulary V_1 than V_2 if V_1 contains more subword tokens from l than V_2. The total number of tokens in l's training data will influence its representation in the vocabulary. Reducing the number of tokens in the training sentences requires increasing the vocabulary capacity. Therefore, fewer tokens in the training data corresponds to a better vocabulary representation. We are interested in comparing BPE and OBPE's vocabulary representation for all languages. We used the following formula to measure the relative change in the number of training tokens when using OBPE instead of BPE,

$$\text{change}_l = \frac{T^l_{\text{OBPE}} - T^l_{\text{BPE}}}{T^l_{\text{BPE}}}\% \quad (1)$$

where T^l_{BPE} and T^l_{OBPE} are the total number of tokens in language l's training data when using BPE and OBPE vocabularies, respectively. Figure 1 shows the change in number of training tokens for all languages. The negative sign in the figure indicates that OBPE represents the language better than BPE. It can be clearly seen that OBPE represents most LRLs better than BPE.

As we are training autoregressive models, the training speed depends on the number of target tokens, which is controlled by the target language representation in the subword vocabulary. Therefore we use the average number of tokens per training example for each language pair (*eng-l*) as a proxy for training speed. Fewer tokens leads to faster training. Both source and target tokens are included, as we are training the model to translate in both directions:

$$\text{AvgTokens}^V_{eng-l} = \frac{\text{Tok}^l_{eng-l} + \text{Tok}^{eng}_{eng-l}}{N_{eng-l}} \quad (2)$$

where $\text{AvgTokens}^V_{eng-l}$ indicates the average number of tokens in one training example from the $eng - l$ dataset using V vocabulary. Tok^l_{eng-l} and Tok^{eng}_{eng-l} represent the total of l and eng tokens, respectively, in the $eng - l$ dataset, while N_{eng-l} represents the number of training examples in the same dataset. Figure 2 shows the average number of training tokens in each language pair when using BPE and OBPE vocabularies. We observe that training with OBPE is slightly faster than training with BPE. The speed difference is higher for languages that are better represented by OBPE (see figure 1).

For these two reasons, and due to time and resources constrains, we chose to continue with training the OBPE multilingual model only.

4.1.2 Bilingual Training

Multilingual models often harm performance on high-resource languages compared to their bilingual counterparts (Yang et al., 2022). For back-translation, we used bilingual models for the subset of language pairs where this happens. We had two translation directions for each language (from/to English) and two vocabulary options (BPE/OBPE) for each direction. We ended up with 32 bilingual models.

All bilingual models were trained on either an `Nvidia` A100 full card (40GB) or a division of half a card (20GB) for 45 epochs with a batch size of 12 288 tokens. The training time depends on the language pairs, but the highest-resource language pair took three days of training.

4.1.3 Back-Translation

For each translation direction, we choose one of the following models for generating back-translation sentences: OBPE bilingual, BPE bilingual, and the 17th epoch checkpoint from the OBPE multilingual model. The selection is based on the models' performance on the Flores dev set, as measured by their BLEU score. We generated the back-translation sentences from the available parallel data only; no additional monolingual data was used. Results from table 4 show the performance of those three models. It can be seen that bilingual models are performing better in both directions of the higher-resource language pairs and for *eng-afr*. We discuss the results in more details in section 5.

We trained the OBPE multilingual model until the 17th epoch. That checkpoint was then used to generate back-translation data for the directions where the multilingual models outperform bilingual ones. Due to resources and time constraints, we started training the back-translation multilingual model from the 17th epoch checkpoint of the OBPE multilingual model. The OBPE multilingual model continued training regularly from the 17th epoch.

We ran all multilingual experiments on 2 `Nvidia` A100 cards (40GB each). One epoch of back-translation or OBPE multilingual models took 16 hours. Both models trained for 45 epochs with a batch size of 16 384 tokens, leading to a total training time of 30 days for each model.

After training both multilingual models, we had four models for each translation direction; two bilingual and two multilingual models.

4.2 Stage 2: Translation Between LRLs

At this stage we found that our models showed adequate performance in the English-centric directions (similar evaluation scores to existing works with overlapping translation directions). The goal of the next stage was to add new translation directions between specific LRLs. Our best multilingual model at this point (based on BLEU scores in the English-centric directions) was the OBPE-based model that was partially trained on back-translated data. Therefore we selected this model to continue training in the new directions. The model trained for an additional 39 epochs on a training set covering the old and new directions (details in section 4.2.2). This took 9 days on a full `Nvidia` A100 card (40GB), at which point validation performance had stopped improving. This resulting model is the system we submitted to the shared task.

4.2.1 Synthetic training data

As shown in table 1, the translation directions between LRLs (new directions) generally had smaller datasets than the directions from/to English (old directions). In fact, two of the new directions (Shona to Afrikaans and Afrikaans to Swati) had no parallel corpora at all. To add these two directions to the model, we generated partially synthetic training data using the available English-centric parallel corpora. Using our multilingual model, we translated the English sentences in the English-Afrikaans corpus to Shona, and the English sentences in the English-Siswati corpus to Afrikaans. This produced parallel corpora for Shona-Afrikaans and Afrikaans-Siswati, where the target sentences were real and the source sentences were synthetic.

4.2.2 Balancing parallel corpora

The challenge in adding new translation directions is to strike a balance between gaining performance in the new directions, while ensuring that performance in the old directions does not deteriorate in the process. For this stage our model was trained on parallel corpora in the old and new directions. Including training data for the old directions ensures that the model does not lose its translation abilities for these directions. However, the parallel corpora for the old directions are on average much larger than those of the new directions. Therefore training on such an unbalanced dataset would likely result in suboptimal performance for new directions.

To counter this, we downsampled the training data for the old directions to match the corresponding corpora in the new directions in order to balance the model's exposure to the old and new directions during training. For example, to balance Xhosa to Zulu training (1M sentences), we trained on 1M sentences only from both the English to Zulu and the Xhosa to English corpora. Therefore the encoder is trained for Xhosa balancing the Xhosa-English and Xhosa-Zulu data, while the decoder is trained for Zulu balancing the English-Zulu and Xhosa-Zulu setting.

Another potentially better approach is upsampling the training data for new directions. This technique would ensure that the model is exposed to all training data of old directions. However, we did not explore this due to time constraints.

5 Results

We primarily used BLEU score for evaluating all models on the Flores dev set. The final test set evaluation by the shared task organizers additionally used sentence piece BLEU (spBLEU) and chrf2.

5.1 Translation Between English and LRLs

Table 4 shows our results on the translation between English and LRLs. For each translation direction, we selected the best model among the two bilingual models and the 17th epoch checkpoint of the OBPE multilingual to perform back-translation. Although the multilingual model was trained only for 17 epochs, it outperformed the fully trained bilingual models in some language pairs. Most of these pairs are resource-poor ($eng \leftrightarrow nso, tso, ssw$). The exception of this finding was the translations between English and Afrikaans. These two languages are from the same family, so we hypothesize that the bilingual models did not need help from other resource-rich pairs or additional training examples to translate between the two languages. The training data of resource-richer language pairs ($eng \leftrightarrow xho, zul, tsn$) were sufficient to train good bilingual models.

After we fully trained both OBPE and OBPE+back-translation multilingual models, the OBPE model performed better than the

Data	Bi-BPE	Bi-OBPE	M-OBPE@17	M-OBPE	M-OBPE+back	M-OBPE-final
sna-eng	19.1	19.6	17.7	19.1	18.1	**19.5**
xho-eng	26.2	26.9	24.2	26.3	26.5	**27.5**
tsn-eng	11.8	11.9	18.1	19.2	16.1	**20.3**
zul-eng	28.7	28.2	26.4	28.6	**30.0**	**30.0**
nso-eng	12.9	14.6	23.1	25.5	22.9	**26.9**
afr-eng	47.5	**48.5**	41.8	45.0	46.4	44.8
tso-eng	1.1	3.3	17.2	18.8	16.9	**20.7**
ssw-eng	0.7	0.9	19.4	21.3	18.0	**23.0**
avg	18.5	19.2	23.5	25.5	24.4	**26.6**
eng-sna	10.1	9.9	9.3	10.0	10.1	**10.3**
eng-xho	12.3	12.6	10.9	11.8	**12.7**	12.1
eng-tsn	10.2	9.6	16.5	17.8	17.8	**18.2**
eng-zul	14.9	14.3	12.6	14.2	**15.1**	15.0
eng-nso	9.8	10.4	20.3	22.1	22.3	**23.1**
eng-afr	**37.2**	35.8	32.3	34.1	36.2	35.6
eng-tso	0.7	0.9	12.8	14.5	15.0	**16.9**
eng-ssw	0.7	0.9	6.2	6.9	7.0	**7.7**
avg	12	11.8	15.1	16.4	17	**17.4**

Table 4: BLEU scores on Flores dev set for translating between English and LRLs. The translation directions are sorted based on the available amount data. Bi-BPE and Bi-OBPE are the BPE and OBPE bilingual models, respectively. M-OBPE@17 is the 17th epoch checkpoints of the OBPE multilingual model, while M-OBPE is trained for 45 epochs. M-OBPE+back and M-OBPE-final are the OBPE with back-translation multilingual models before and after continued training for translation between LRL, respectively. underline indicates the model we used for back-translation. **Bold** represents the best overall model.

Data	M-OBPE+back	M-OBPE-final
xho-zul	1.5	**11.2**
zul-sna	1.9	**8.8**
sna-afr	1.9	**12.2**
afr-ssw	1.3	**4.9**
ssw-tsn	2.0	**14.5**
tsn-tso	2.1	**13.6**
tso-nso	2.4	**13.2**
nso-xho	1.7	**8.2**
avg	1.8	**10.8**

Table 5: BLEU scores on Flores dev set for translating between LRLs. M-OBPE+back and M-OBPE-final are the OBPE multilingual models with back-translation before and after continued training for translation between LRL, respectively. M-OBPE-final is the system we submitted for the shared task. **Bold** represents the best results.

back-translation model in most directions with English as a target language, namely, $sna, tsn, nso, tso, ssw \rightarrow eng$. However, for the three eng generation directions where the back-translation model performed similarly or better than the OBPE model ($xho, zul, afr \rightarrow eng$), the back-translation data was generated from the bilingual models, not the OBPE multilingual model. This synthetic data contains actual English sentences and synthetic LRLs sentences. These translation pairs were relatively resource-rich. In contrast, most of the remaining pairs were resource-poor, and their back-translation data was generated from the partially trained OBPE multilingual model. These results show that although the 17th epoch checkpoint of the OBPE multilingual model was better than bilingual models in resource-poor language pairs, it was not yet good enough for generating text in LRLs. This led to a performance drop for the back-translation model on most of the eng generation directions compared to the OBPE multilingual model.

On the other hand, the back-translation model outperformed the OBPE model in all directions translating into LRLs. These directions require synthetic English sentences and actual LRLs sentences for back-translation. A plausible explanation for this is that learning to translate to English is easier than translating to LRLs for both bilingual and multilingual models.

Data	BLEU	spBLEU	CHRF2++	ΔCHRF2++
sna-eng	18.7	22.1	42.9	5.5
xho-eng	24.3	26.8	47.7	5.6
tsn-eng	19.8	22.1	42.6	7.7
zul-eng	26.7	28.5	49.3	6.5
nso-eng	26.5	28	48.1	9.4
afr-eng	44.7	46.4	66	9
tso-eng	20.3	21.8	41.9	8.8
ssw-eng	21.5	23.5	43.8	7.9
avg	25.31	27.4	47.79	7.55
eng-sna	10.3	17.6	41.1	2.9
eng-xho	9.4	18.6	42.5	3.4
eng-tsn	18.8	19.7	43	5
eng-zul	11.9	22.8	46.1	3.4
eng-nso	22.7	24.1	47.8	4
eng-afr	35.9	40.5	62.2	3.6
eng-tso	15.8	17.9	41.5	4.8
eng-ssw	7.6	15.5	38.9	4.4
avg	16.55	22.09	45.39	3.94
xho-zul	8.5	18	41.4	1.9
zul-sna	8.5	15	38.7	1.7
sna-afr	12	15.1	38	3.9
afr-ssw	5.3	11.2	34.3	7.2
ssw-tsn	14.4	15.4	38.9	2.9
tsn-tso	13.2	15.1	38.7	2.1
tso-nso	13.1	12	36.6	5.8
nso-xho	6.6	13.7	36.9	4
avg	10.2	14.44	37.94	3.69
overall avg	17.35	21.31	43.7	5.06

Table 6: The performance of our final system on the shared task test set. Δ CHRF2++ is the difference between the best submission and our system.

5.2 Translation between LRLs

Table 5 shows the performance of the OBPE+back-translation model before and after continued training for translation between LRLs. The model's performance improved on both the initial language pairs (in table 4) and the new translation directions. Moreover, $sna \rightarrow afr$ and $afr \rightarrow ssw$ were improved using only synthetic data (see section 4.2.1). We ascribe the success in improving the model's performance in translating between English and LRLs to the balancing approach (see section 4.2.2), as we used real training data (not back-translated sentences) in the continued training.

5.3 Official Results

Table 6 shows the results provided by the shared task organizers for our system as evaluated on a hidden test set. The table also compares the best constrained submission for each translation direction and our system. Our model did not achieve the best performance in any direction. However, the teams whose models performed better all trained on all languages included in the shared task (not just Southern African languages).

We hypothesize that this is the main reason for the gap in performance between our system and the better performing ones, as those models could benefit from more training data and increased cross-lingual transfer. The fact that our model performs relatively worse when translating into English provides some evidence for this: the other systems could benefit learning to translate to English in many more translation directions and with much more data in total. Given our computational resources, it would have required a total training time of 106 days to cover all language directions in the shared task. Unfortunately this was not feasible in the time provided for the shared task. The findings paper for the shared task presents more details about other teams' submissions (Adelani et al., 2022).

6 Conclusion

We have presented our multilingual neural MT model for 8 Southern African languages. Until recently, it would not have been possible to train a multilingual model for these languages because of data scarcity. During model development we found the benefits of multilingual modelling to be especially great for the lowest-resourced languages. Our results show that overlap BPE, back-translation, and synthetic training data generation are all valuable techniques for low-resource MT. More generally, we find multilingual modelling to be a fruitful approach to Southern African MT. For future work we would like to investigate further approaches for training large multilingual models for low-resource languages with a limited compute budget.

Acknowledgements

This work is based on research supported in part by the National Research Foundation of South Africa (Grant Number: 129850). Computations were performed using facilities provided by the University of Cape Town's ICTS High Performance Computing team: `hpc.uct.ac.za`. Francois Meyer is supported by the Hasso Plattner Institute for Digital Engineering, through the HPI Research School at the University of Cape Town.

References

David Ifeoluwa Adelani, Md Mahfuz Ibn Alam, Antonios Anastasopoulos, Akshita Bhagia, Marta Costa-

Jussá, Jesse Dodge, Fahim Faisal, Christian Federmann, Natalia Fedorova, Francisco Guzmán, Sergey Koshelev, Jean Maillard, Vukosi Marivate, Jonathan Mbuya, Safiyyah Saleem, and Holger Schwenk. 2022. Findings of the WMT 2022 shared task on large-scale machine translation evaluation for african languages. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation.

Yunsu Kim, Yingbo Gao, and Hermann Ney. 2019. Effective cross-lingual transfer of neural machine translation models without shared vocabularies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1246–1257, Florence, Italy. Association for Computational Linguistics.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Auguste Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara E. Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Javier Ortiz Suárez, Iroro Fred Onome'Orife, Kelechi Ogueji, Rubungo Andre Niyongabo, Toan Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2021. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics (TACL)*.

Baohao Liao, Shahram Khadivi, and Sanjika Hewavitharana. 2021. Back-translation for large-scale multilingual machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 418–424, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Vaidehi Patil, Partha Talukdar, and Sunita Sarawagi. 2022. Overlap-based vocabulary generation improves cross-lingual transfer among related languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 219–233, Dublin, Ireland. Association for Computational Linguistics.

Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual adapters for zero-shot neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4465–4470, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Jian Yang, Yuwei Yin, Shuming Ma, Dongdong Zhang, Zhoujun Li, and Furu Wei. 2022. High-resource language-specific training for multilingual neural machine translation. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4461–4467. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Tencent's Multilingual Machine Translation System for WMT22 Large-Scale African Languages

Wenxiang Jiao Zhaopeng Tu Jiarui Li Wenxuan Wang Jen-tse Huang Shuming Shi
Tencent AI Lab
{joelwxjiao,zptu,ultrali,jwxwang,jentsehuang,shumingshi}@tencent.com

Abstract

This paper describes Tencent's multilingual machine translation systems for the WMT22 shared task on Large-Scale Machine Translation Evaluation for African Languages. We participated in the **constrained** translation track in which only the data and pretrained models provided by the organizer are allowed. The task is challenging due to three problems, including the absence of training data for some to-be-evaluated language pairs, the uneven optimization of language pairs caused by data imbalance, and the curse of multilinguality. To address these problems, we adopt data augmentation, distributionally robust optimization, and language family grouping, respectively, to develop our multilingual neural machine translation (MNMT) models. Our submissions won the **1st** place on the blind test sets in terms of the automatic evaluation metrics.[1]

1 Introduction

Multilingual neural machine translation (MNMT) aims to translate between multiple language pairs with a unified model (Johnson et al., 2017). It is appealing due to the model efficiency, easy deployment, and knowledge transfer between high resource languages and low resource languages. Hence, MNMT has attracted more and more attention from both academia and industry. To improve the performance of MNMT models, previous researchers have proposed various approaches on advanced model architectures (Sen et al., 2019; Zhang et al., 2021), training strategies (Wang et al., 2020a,b), and data utilization (Siddhant et al., 2020; Wang et al., 2022). In addition, industrial companies have released massive multilingual pretrained models (Tang et al., 2021) and large-scale multilingual translation models (Fan et al., 2021; Team et al., 2022) to facilitate translation among hundreds of languages. However, existing efforts on MNMT for African languages are not sufficient due to the lack of high quality and standardized evaluation benchmarks.

In this paper, we build a system integrating several advanced approaches for WMT22 Large-Scale Machine Translation Evaluation Task (Adelani et al., 2022), which involves a set of 24 African languages. We participated in the Constrained Translation track, where only the data provided by the organizer are allowed. This task is challenging due to three potential problems:

- The absence of training data for some to-be-evaluated language pairs;
- The uneven optimization of language pairs due to data imbalance;
- The curse of multilinguality in MNMT models caused by the hundreds of language pairs.

For the first problem, we adopt data augmentation techniques to construct synthetic data for the language pairs without parallel training data (§3.1). Specifically, we use back-translation (Sennrich et al., 2016) and self-training (Jiao et al., 2021), and attach a special tag to the synthetic side of the data. For the second issue, we utilize distributionally robust optimization (DRO) method (Oren et al., 2019; Zhou et al., 2021) to balance the optimization process for different translation directions (§3.2). For the third issue, we isolate the potential conflicts between language pairs by language family grouping and finetune a model for each language group (§3.3).

Experimental results show that our system can significantly improve the performance of vanilla MNMT models, from 15.50 to 17.95 BLEU points (§4.2). Extensive analysis suggests that data augmentation could be harmful to the translation performance if used for training the final models

[1] Codes, models, and detailed competition results are available at https://github.com/wxjiao/WMT2022-Large-Scale-African.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1049 - 1056
December 7-8, 2022 ©2022 Association for Computational Linguistics

Table 1: Information of language groups and the corresponding language pairs. We include additional 36 language pairs (**bolded**) to help the long-tail languages.

Group	Language Pairs	(73) **(117)**
ENGC	afr-eng,amh-eng,eng-fra,eng-fuv,eng-hau,eng-ibo,eng-kam,eng-kin,eng-lug,eng-luo,eng-nso,eng-nya,eng-orm,eng-sna,eng-som,eng-ssw,eng-swh,eng-tsn,eng-tso,eng-umb,eng-xho,eng-yor,eng-zul, (23) ,**eng-lin, eng-wol, (25)**	
FRAC	fra-kin,fra-lin,fra-swh,fra-wol, (4) ,**amh-fra,fra-kam,fra-lug,fra-luo,fra-orm,fra-umb, (10)**	
SSEA	**afr-nso**,**afr-sna**,afr-ssw,afr-tsn,afr-xho,afr-tso,afr-zul,nso-sna,nso-ssw,nso-tsn,nso-xho,nso-tso,nso-zul,sna-ssw,sna-tsn,sna-xho,sna-tso,sna-zul,ssw-tsn,ssw-xho,ssw-tso,ssw-zul,tsn-xho,tsn-tso,tsn-zul,tso-xho,tso-zul,xho-zul, (28)	
HCEA	**amh-luo**,amh-orm,amh-som,amh-swh,luo-orm,luo-som,luo-swh,orm-som,orm-swh,som-swh, (10)	
NGG	fuv-hau,fuv-ibo,fuv-yor,hau-ibo,hau-yor,ibo-yor, (6)	
CA	**kin-lin**,kin-lug,kin-nya,kin-swh, **lin-lug**,**lin-nya**,**lin-swh**,lug-nya,lug-swh,nya-swh, (10)	
OTHER	**fuv-kin,fuv-nya,fuv-som,fuv-zul,kam-nya,kam-sna,kam-som,kam-swh,kam-tso,kam-zul,kin-yor,lug-sna,lug-zul,luo-nya,luo-sna,luo-zul,nya-umb,nya-yor,sna-umb,sna-yor,som-wol,som-yor,swh-umb,swh-yor,tso-yor,umb-zul,xho-yor,yor-zul, (28)**	

directly, due to the error-prone synthetic sentence pairs. Instead, we utilize the resulting MNMT models as pretrained models to further finetune on clean datasets for the final models. The DRO technique is very effective in improving the translation quality across all language pairs, particularly on the dominant languages (e.g., eng and fra), which also calls for an improved DRO to benefit more on other languages. As for language family grouping, it especially improves the translation quality on one-to-many translations, which demonstrates its effectiveness in alleviating the curse of multilinguality issue. Finally, our submission won the **1st** place in the official evaluation in terms of the automatic evaluation metrics.

2 Data

In this section, we present the details of our data preparation.

2.1 Language Pairs

We utilize all available datasets from the official website (including those from the Data Track participants)[2], which provide either monolingual or parallel sentences. According to the evaluation instruction, we group the language pairs into 7 groups, namely, English-Centric (ENGC), French-Centric (FRAC), South/South East Africa (SSEA), Horn of Africa and Central/East Africa (HCEA), Nigeria and Gulf of Guinea (NGG), Central Africa (CA), and Other related pairs (OTHER), to train the MNMT models. Details are listed in Table 1.

We consider three subsets of language pairs for training different models:

- **Base-146**: We train the TRANSF-DEEP (§4.1) models on the to-be-evaluated language pairs in the first 6 groups, as well as the English-French (i.e., eng-fra) pair. In total, there are 81 language pairs but only 73 of them are provided with bitext data, which cover 146 translation directions (i.e., including both forward and backward).

- **Large-234**: The main issues of **Base-146** are that, some to-be-evaluated language pairs (e.g., afr-nso) are missing in the training data and some languages are heavily long-tailed due to the imbalanced choice of language pairs. To alleviate these issues, we extend another 36 language pairs for the long-tail languages and construct synthetic data for all the language pairs in ENGC, SSEA, HCEA, NGG and CA, which enables the training on 234 translation directions. We use these language pairs to train the TRANSF-DWIDE (§4.1) models.

- **Eval-106**: The official evaluation includes 100 translation directions[3], which were notified at the later stage of the competition. We focus on these directions by finetuning the TRANSF-DWIDE models on these directions. To ensure the data amount of each language, we include all ENGC directions, making the final 106 directions.

[2] https://www.statmt.org/wmt22/large-scale-multilingual-translation-task.html

[3] https://docs.google.com/document/d/11NYyJpJ4nhNIllwmF5kjkqfkaaEzXNU-CCO5E64MRDU/edit

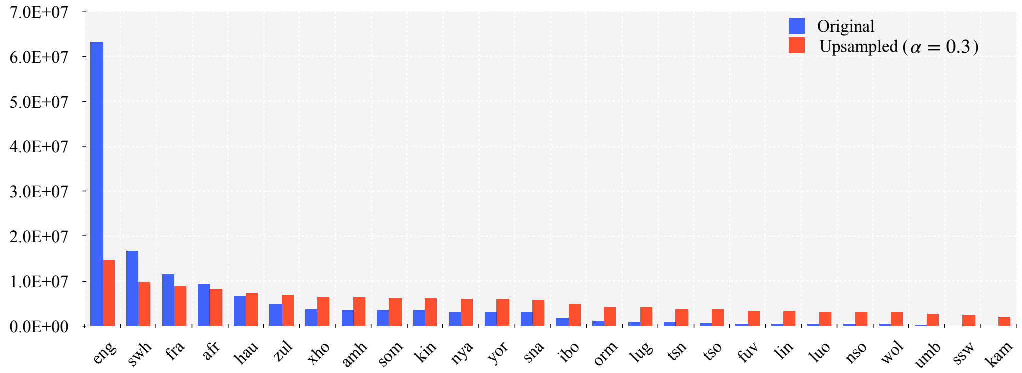


Figure 1: Number of sentences in each language and the upsampled distribution with the smoothing rate of $\alpha = 0.3$.

2.2 Data Preprocessing

We preprocess the raw and potentially noisy data by four steps, namely, reformatting, deduplication, language detection, and length limitation. Details are elaborated as below.

Reformatting. The raw data is stored in various alignment structures, including HTML, JSON, and special spacing. To reduce data noise, we reformat all data into a line-by-line tight structure and realign those missing paired ones.

Deduplication. We remove the duplicated sentences (pairs) in each monolingual and parallel dataset. This aims to reduce information redundancy so that the MNMT models can be trained more efficiently.

Language Detection. Previous studies suggest that incorrect languages in training data induce translation uncertainty for both bilingual (Ott et al., 2018) and multilingual (Wang et al., 2022) NMT models. Therefore, we conduct language detection for all the datasets using `langid`[4]. Since `langid` neither supports all African languages nor performs well when distinguishing two African languages, we adopt a simplified strategy: for the African datasets, we remove sentences (pairs) that are identified as languages other than the 24 designated African languages. In other words, sentences (pairs) in one African language identified as another African language are also considered valid. For English and French datasets, we strictly restrict the correct languages as themselves, i.e., English and French, respectively.

Length Limitation. After multilingual tokenization, we conduct further filtering and retain sentence pairs with tokens between 4 (Wu et al., 2019) and 512 (Yang et al., 2021), as well as the length ratio below 3.

2.3 Multilingual Tokenization

To tokenize the multilingual sentences, we follow (Conneau et al., 2020) to train a Sentence Piece Model (SPM) and apply it directly on the preprocessed text data for all languages. However, the distribution of data across languages is heavily long-tailed, as shown in Figure 1. To balance the vocabulary bandwidth between high-resource and low-resource languages, we follow Conneau et al. (2020) to upsample the low-resource languages with a smoothing rate of $\alpha = 0.3$ over the original distribution when training the SPM model. We use a shared vocabulary with 128K tokens for the 26 languages, and also append 32 special tokens (i.e., "TBD0" to "TBD31") for including extra tasks or data (e.g., tagged-BT (Caswell et al., 2019)).

3 Approach

3.1 Data Augmentation

We adopt data augmentation (DA) to address the first challenge, i.e., " **The absence of training data for some to-be-evaluated language pairs**".

Specifically, we use back-translation (BT) (Sennrich et al., 2016) and self-training (ST) (Jiao et al., 2020, 2021, 2022) to construct synthetic data. However, previous study by Caswell et al. (2019) suggests that the translationese issue in BT limits the performance, which can be mitigated with a special tag at source side (i.e., tagged-BT). To simplify the tagging procedure for the two opposite directions of each language pair, we use both BT and ST for each language pair (Wu et al., 2019) and append a special token at the synthetic side of sentence pairs. Formerly, for a language pair (S, T) with the bitext data $\{\mathbf{x}, \mathbf{y}\}$, the synthetic data by BT and ST will

[4] `https://github.com/saffsd/langid.py`

be $\{[\mathbf{x}'; \langle \texttt{DA} \rangle], \mathbf{y}\}$ and $\{\mathbf{x}, [\mathbf{y}'; \langle \texttt{DA} \rangle]\}$, where $\langle \texttt{DA} \rangle$ denotes the special tag for data augmentation.

We conduct data augmentation for both English-centric and non English-centric language pairs. For English-centric language pairs, we randomly sample up to 1.0M English and non-English monolingual sentences from the training corpora for BT and ST, respectively. As for non English-centric language pairs, we translate the English side of English-centric pairs to non-English languages and construct up to 0.5M BT and ST sentence pairs, respectively. Generally, the augmented data is included in **Large-234** to train the MNMT models. However, the translation quality of those English-centric directions is also unreliable due to the limited data sizes, which may harm the performance of subsequent MNMT models. Besides, adding more synthetic data and language directions also slows down the convergence of the MNMT models. Instead, we use the resulting MNMT models as backbones to finetune on the clean datasets.

3.2 Distributionally Robust Optimization

We adopt the distributionally robust optimization (DRO) (Oren et al., 2019; Zhou et al., 2021) technique to address the second challenge, i.e., "**The uneven optimization of language pairs due to data imbalance** ".

Generally, temperature-based sampling (Arivazhagan et al., 2019; Conneau et al., 2020) is adopted to balance the training data across language pairs, which samples data from the smoothed data distribution as, $p_{\tau,i} = \frac{|D_i|^{1/\tau}}{\sum_j |D_j|^{1/\tau}}$. This is equivalent to optimizing the re-weighted objective:

$$\mathcal{L}_\tau(\theta; D_{\text{train}}) = \sum_{i \le N} p_{\tau,i} \mathcal{L}(\theta; D_i), \tag{1}$$

where $|D_i|$ is the training data size of the i-th language pair, and τ denotes the temperature rate. Obviously, $\tau = 1$ corresponds to the original data distribution while $\tau = \infty$ represents uniform sampling. In practice, $\tau > 1$ is adopted to oversample the low-resource language pairs, which significantly affects the results and needs to be tuned for different settings.

Even if we can build a completely balanced dataset across language pairs, the varied task difficulty and cross-lingual similarity determine that the language pairs will still be optimized unevenly. DRO can address such a problem. In contrast to temperature sampling which optimizes over a fixed training data distribution, DRO aims to find a model θ that can perform well on an entire set of potential test distributions, i.e., $\mathcal{U}(p^{\text{train}})$, which is usually called *uncertainty set*. We adopt DRO with the χ^2-uncertainty set introduced by Zhou et al. (2021), and reproduce the implementation for the practical many-to-many translation scenario.[5] Similarly, we also incorporate the baseline losses calculated from a pretrained MNMT model to stabilize the training process of DRO.

Table 2: Language family grouping.

Group	Target Languages
1	eng, fra
2	afr, nso, sna, ssw, tsn, tso, xho, zul
3	amh, luo, orm, som, swh, wol
4	fuv, hau, ibo, yor
5	kam, kin, lin, lug, nya, swh, umb

3.3 Language Family Grouping

We adopt language family grouping (LFG) to alleviate the third challenge, i.e., " **The curse of multilinguality**" (Conneau et al., 2020).

Specifically, we divide the target languages into 5 groups (see Table 2) based on Table 1. This is partially inspired by Eriguchi et al. (2022), which factorizes the many-to-many translation scenario (with $N \times N$ directions) into N many-to-one scenarios by training a translation model for each. Since we have 26 languages involved in this shared task, factorizing the many-to-many scenario by the family of target languages is a more efficient choice. Since **swh** appears in both HCEA and CA, we include it in both Group-2 and Group-5 for training models. During inference, our scripts will automatically select the model of corresponding group according to the target language to be evaluated. Note that **swh** is only routed to Group-2 in inference.

4 Experiments

4.1 Settings

Model. We adopt the standard sequence-to-sequence Transformer (Vaswani et al., 2017) as our architecture. For the **Base-146** scale, we use a deep encoder of 24 layers and a relatively shallow decoder of 12 layers (Yang et al., 2021), with an embedding size of 1024, the feed-forward network

[5] The referred study only supports one-to-many and many-to-one translation scenarios on very small multilingual translation datasets.

Table 3: Evaluation results of our models on the devtest in terms of BLEU and ChrF++.

Model	X-Eng	Eng-X	X-Fra	Fra-X	X-X	All
	22	22	4	4	48	100
Transf-Deep	23.37/46.80	17.19/41.07	20.20/43.18	16.07/41.93	10.69/33.90	15.50/39.01
Borderline-Deep	25.87/48.83	18.24/42.05	22.31/44.91	16.74/42.32	11.66/34.89	16.86/40.23
Borderline-DWide	28.11/51.30	19.02/42.92	24.74/47.58	17.22/43.23	12.03/34.94	17.82/41.13
Borderline-DWide w/ LFG	28.26/51.37	19.38/43.37	24.87/47.74	17.48/43.72	12.04/35.01	**17.95/41.31**

size of 4096, and 16 attention heads (i.e., 0.59B parameters). To stabilize the training of deep models, we follow Wang et al. (2019) to use pre-layer-normalization (PLN) for both encoder and decoder layers. For the **Large-234** scale, we enlarge the embedding size to 1536 to support more language pairs, which results in 1.02B parameters. By default, we call these two models as Transf-Deep and Transf-DWide. The final models developed by our approaches are renamed as Borderline-Deep and Borderline-DWide for clarity.

Training. We train the MNMT models with the Adam optimizer (Kingma and Ba, 2014) ($\beta_1 = 0.9, \beta_2 = 0.98$). The learning rate is set as 1e-4 with a warm-up step of 4000, followed by inverse square root decay. The models are trained with a dropout rate of 0.1 and a label smoothing rate of 0.1. All experiments are conducted on 32 NVIDIA A100 GPUs. Since the bitext data ($\approx$130M) for this year's shared task is less than 1/10 of that for last year's ($\approx$1.7B), we decide a batch size to be about 1/10 of that used in (Yang et al., 2021). Specifically, we use 2048 max-tokens per GPUs and accumulate the gradients for every 8 steps to simulate the large batch size of 512K tokens. For language family grouping, we use the batch size of 131K tokens for each model. For translation models trained by empirical risk minimization (ERM) on the original training data, we upsample low-resource language pairs with the smoothing rate $\alpha = 0.3$ (Conneau et al., 2020). For those by DRO, we adopt the χ^2-uncertainty set with the distribution divergence bounded by $\rho = 0.1$. We use the ERM model to calculate the baseline losses for DRO. We train these two kinds of models for at least 100K updates, upon which we may finetune for additional updates.

Evaluation. We use the dev and devtest of Flores-200 benchmark[6] as our validation and test sets, and evaluate the MNMT models on the averaged last 10 checkpoints with sentencepiece BLEU and ChrF++. The sentencepiece model for evaluation also comes from the Flores-200 benchmark. The beam search process is performed with a beam size of 4 and a length penalty of 1.0. Similar as the official competition results, we report our results by average-to-eng (X-Eng), average-from-eng (Eng-X), average-to-fra (X-Fra), average-from-fra (Fra-X), average-african-to-african (X-X), and the average for All translation directions.

4.2 Results

We list the evaluation results of our final models on the devtest in Table 3. Both the baseline model Transf-Deep and our Borderline-Deep model are trained for 200K updates, while the two Borderline-DWide models are trained or finetuned for more than 300K updates.

Generally, our models outperform the baseline Transf-Deep model significantly by up to +2.45 BLEU and +2.30 ChrF++ scores. By looking into each category, we have some interesting findings:

- By comparing Borderline-Deep and Transf-Deep, we find that the improvement on X-Eng is much larger than that on Eng-X. Similar phenomenon is also observed for X-Fra and Fra-X. It suggests that while DRO can achieve even improvement for one-to-many or many-to-one scenarios (Zhou et al., 2021), it is heavily biased by the dominant languages (i.e., eng and fra) in the many-to-many scenario.

- By comparing Borderline-DWide and Borderline-Deep, we find that enlarging the model capacity brings improvement to all categories but the most on X-Eng and X-Fra. It indicates that the *curse of multilinguality* cannot be well solved by simply increasing model capacity as the most benefits are still occupied by the dominant languages (i.e., eng and fra).

- Language family grouping (LFG) achieves more improvement on Eng-X and Fra-X than on the

[6] https://github.com/facebookresearch/flores/tree/main/flores200

Table 4: Official evaluation results of submissions on the blind test sets in terms of BLEU and ChrF++.

Submissions	X-Eng	Eng-X	X-Fra	Fra-X	X-X	All
#Lang-pairs	22	22	4	4	48	100
				IIAI		
Primary	23.15/43.88	12.80/37.52	18.35/41.08	13.08/38.70	2.58/19.52	10.40/30.47
				GMU		
Language	25.83/46.50	12.00/35.33	20.83/42.45	10.53/33.58	7.70/29.94	13.28/35.42
Family	25.88/46.55	11.98/35.30	20.73/42.30	10.75/34.03	7.68/29.92	13.28/35.42
				Borderline (Ours)		
Contrastive	25.84/47.46	13.85/39.05	21.00/44.10	13.85/39.58	8.03/30.93	13.98/37.23
Primary	26.05/47.56	14.06/39.53	21.13/44.05	14.05/40.10	8.04/31.04	**14.09/37.42**

Table 5: Ablation study of our models with various strategies on the devtest. CT: continuous training; FT: finetuning; T-Enc: target language tags at encoder; LFG: language family grouping.

ID	Model	Step	BLEU	Δ
①	Transf-Deep	100K	15.03	-/-
②	+ CT	100K	15.50	+0.47
③	+ FT on large-234	100K	14.65	-0.38
④	+ DRO	100K	16.71	+1.68
⑤	+ CT	100K	16.86	**+1.83**
⑥	+ T-Enc	100K	16.67	+1.64
⑦	Transf-DWide	100K	14.66	-/-
⑧	+ DRO	100K	15.81	+1.15
⑨	+ FT on base-146	200K	17.62	+2.96
⑩	+ FT on eval-106	50K	17.82	+3.16
⑪	+ LFG	-/-	17.95	+3.29

other categories, which confirms its effectiveness in alleviating the curse of multilingualty issue.

Ablation Study. We present detailed ablation studies to investigate the effectiveness of various strategies, not only the three introduced in §3 but also some tricks. The results are listed in Table 5, where the lines marked in blue (i.e., ②, ⑤, ⑩ and ⑪) correspond to the four models in Table 3. We list our observations as below:

- ③ *vs.* ②: Directly finetuning the Transf-Deep model on the **Large-234** dataset induces the performance drop. One possible reason is that **Large-234** introduces much more translation directions, aggravating the *curse of multilinguality* issue. Another reason is the low-quality data by data augmentation (§3.1), which harms the optimization of models. Therefore, we only use **Large-234** to pretrain the Transf-DWide model and then finetune on the cleaner **Base-146** and **Eval-106** datasets.

- ⑥ *vs.* ⑤: Previous studies (Wang et al., 2022) suggest that attaching target language tags at encoder (i.e., T-Enc) benefits the zero-shot translation performance, indicating a stronger cross-lingual transfer ability. However, we do not see any improvement of our models with T-Enc. The reason could be that, traditional studies on many-to-many translations are mainly conducted on the datasets with only one central language while we are now handling multiple central languages, making it a more complex scenario.

- ⑨ *vs.* ⑩ *vs.* ⑪: Finetuning on **Eval-106** slightly outperforms that on **Base-146** and the performance can be further improved with language family grouping. Obviously, as we reduce the language pairs involved in a single model, the *curse of multilinguality* is alleviated.

Submissions. The Borderline-DWide and Borderline-DWide w/ LFG models shown in Table 3 (i.e., contrastive and primary versions) are submitted for official evaluation on the blind test sets. Table 4 summarizes the evaluation results of our submissions, where our models outperform the other teams' across all the evaluation groups. Finally, we achieve the **1st** place in this track.

5 Conclusion

In this paper, we describe Tencent's multilingual machine translation systems for the WMT22 shared task on Large-Scale Machine Translation Evaluation for African Languages. We address three key challenges of this task by data augmentation, distributionally robust optimization (DRO), and language family grouping, respectively, to develop our MNMT models. Our submissions won the **1st** place in the **constrained** track. Extensive analyses also point out the drawbacks of larger models and DRO in addressing the curse of multilinguality, which warrants further research in the future.

Acknowledgments

We sincerely thank the evaluation group for their hard work, including but not limited to the instant communication, environment configuration, and official evaluation. We also thank the anonymous reviewers for their insightful suggestions on various aspects of this report.

References

David Ifeoluwa Adelani, Md Mahfuz Ibn Alam, Antonios Anastasopoulos, Akshita Bhagia, Marta Costa-Jussá, Jesse Dodge, Fahim Faisal, Christian Federmann, Natalia Fedorova, Francisco Guzmán, Sergey Koshelev, Jean Maillard, Vukosi Marivate, Jonathan Mbuya, Safiyyah Saleem, and Holger Schwenk. 2022. Findings of the WMT 2022 shared task on large-scale machine translation evaluation for african languages. In *WMT*.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv*.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *WMT*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *ACL*.

Akiko Eriguchi, Shufang Xie, Tao Qin, and Hany Hassan Awadalla. 2022. Building multilingual machine translation systems that serve arbitrary xy translations. In *NAACL*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Çelebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*

Wenxiang Jiao, Xing Wang, Shilin He, Irwin King, Michael Lyu, and Zhaopeng Tu. 2020. Data rejuvenation: Exploiting inactive training examples for neural machine translation. In *EMNLP*.

Wenxiang Jiao, Xing Wang, Shilin He, Zhaopeng Tu, Irwin King, and Michael R Lyu. 2022. Exploiting inactive examples for natural language generation with data rejuvenation. *IEEE/ACM TASLP*.

Wenxiang Jiao, Xing Wang, Zhaopeng Tu, Shuming Shi, Michael Lyu, and Irwin King. 2021. Self-training sampling with monolingual data uncertainty for neural machine translation. In *ACL*.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Z. Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv*.

Yonatan Oren, Shiori Sagawa, Tatsunori B Hashimoto, and Percy Liang. 2019. Distributionally robust language modeling. In *EMNLP-IJCNLP*.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *ICML*.

Sukanta Sen, Kamal Kumar Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2019. Multilingual unsupervised nmt using shared encoder and language-specific decoders. In *ACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *ACL*.

Aditya Siddhant, Ankur Bapna, Yuan Cao, Orhan Firat, Mia Xu Chen, Sneha Kudugunta, N. Arivazhagan, and Yonghui Wu. 2020. Leveraging monolingual data with self-supervision for multilingual neural machine translation. *ACL*.

Y. Tang, C. Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *FINDINGS*.

Nllb Team, Marta Ruiz Costa-jussà, James Cross, Onur cCelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Alison Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon L. Spruit, C. Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzm'an, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *ArXiv*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. 2019. Learning deep transformer models for machine translation. In *ACL*.

Wenxuan Wang, Wenxiang Jiao, Shuo Wang, Zhaopeng Tu, and Michael R Lyu. 2022. Understanding and mitigating the uncertainty in zero-shot translation. *arXiv*.

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020a. Balancing training for multilingual neural machine translation. *ACL*.

Yiren Wang, ChengXiang Zhai, and Hany Hassan Awadalla. 2020b. Multi-task learning for multilingual neural machine translation. *EMNLP*.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *EMNLP-IJCNLP*.

Jian Yang, Shuming Ma, Haoyang Huang, Dongdong Zhang, Li Dong, Shaohan Huang, Alexandre Muzio, Saksham Singhal, Hany Hassan, Xia Song, et al. 2021. Multilingual machine translation systems from microsoft for wmt21 shared task. In *WMT*.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021. Share or not? learning to schedule language-specific capacity for multilingual translation. In *ICLR*.

Chunting Zhou, Daniel Levy, Xian Li, Marjan Ghazvininejad, and Graham Neubig. 2021. Distributionally robust multilingual machine translation. In *EMNLP*.

DENTRA: Denoising and Translation Pre-training for Multilingual Machine Translation

Samta Kamboj, Sunil Kumar Sahu, Neha Sengupta
Inception Institute of Artificial Intelligence
Abu Dhabi, UAE
samta.kamboj@g42.ai, {sunil.sahu, neha.sengupta}@inceptioniai.org

Abstract

In this paper, we describe our submission to the *WMT-2022: Large-Scale Machine Translation Evaluation for African Languages* under the *Constrained Translation track*. We introduce DENTRA, a novel pre-training strategy for a multilingual sequence-to-sequence transformer model. DENTRA pre-training combines denoising and translation objectives to incorporate both monolingual and bitext corpora in 24 African, English, and French languages. To evaluate the quality of DENTRA, we fine-tuned it with two multilingual machine translation configurations, one-to-many and many-to-one. In both pre-training and fine-tuning, we employ only the datasets provided by the organisers. We compare DENTRA against a strong baseline, M2M-100, in different African multilingual machine translation scenarios and show gains in 3 out of 4 subtasks.

1 Introduction

Despite the compelling performance of machine translation (MT) in many European and Asian languages, their quality in African languages is relatively low. This is primarily because there are approximately 2000 known languages in the African continent, out of which very few languages have any significant presence on the Web (Eberhard et al., 2020; Emezue and Dossou, 2021; Adelani et al., 2022a). As a result, many African languages are not included in publicly available bitext resources, which are typically created by employing heuristics on large amounts of data crawled from the Web (Tiedemann, 2012; El-Kishky et al., 2020; Schwenk et al., 2021; Goyal et al., 2022).

To take a step towards addressing the underrepresentation of African languages in MT, WMT-2022 presented the Constrained Translation track under *Large-Scale Multilingual African Translation* (Adelani et al., 2022b), which releases bitext and monolingual corpora for 24 African languages, and participants are only allowed to use the provided data. Our submission is to the aforementioned track.

Roughly 34% of the provided data is monolingual, spread across 24 African languages pertaining to this task. Since the volume of bitext data provided is limited, our submission aims to leverage the monolingual data to improve the performance of a multilingual machine translation model. To leverage monolingual data in translation, pretraining the model is an obvious choice.

There are several existing multilingual pretrained models such as mBART (Liu et al., 2020), mT5 (Xue et al., 2021), byT5 (Xue et al., 2022), mRASP (Pan et al., 2021), mRASP2 (Pan et al., 2021), and M2M-100[1] (Fan et al., 2021) that are trained on monolingual data, bitext data, or both, and have been demonstrated to improve translation performance for specific language pairs. But, these models do not include many of the African languages of interest. For example, the 50 languages covered by mBART50 include only two out of the 24 African languages in the shared task while M2M-100 includes 14. Moreover, all of these multilingual models rely on specially designated language id tokens to translate between each language pair. As a result, adding unseen languages requires pre-training again. Adelani et al. (2022a) investigated a way to leverage pre-trained models including M2M-100, mT5, byT5, and mBART for the translation of unseen languages. But the scarcity of African language texts in the pre-training corpora results in a marginal improvement in the translation quality of the fine-tuned model (Adelani et al., 2022a). Among the pre-trained models, the authors have noted that fine-tuning M2M-100 results in the best translation performance for African languages.

[1] Although M2M-100 is trained for many-to-many translation tasks, it has been used as a pre-trained model by Adelani et al. (2022a) for African MT. Therefore, we also consider it as a pre-trained model in this work.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1057 - 1067
December 7-8, 2022 ©2022 Association for Computational Linguistics

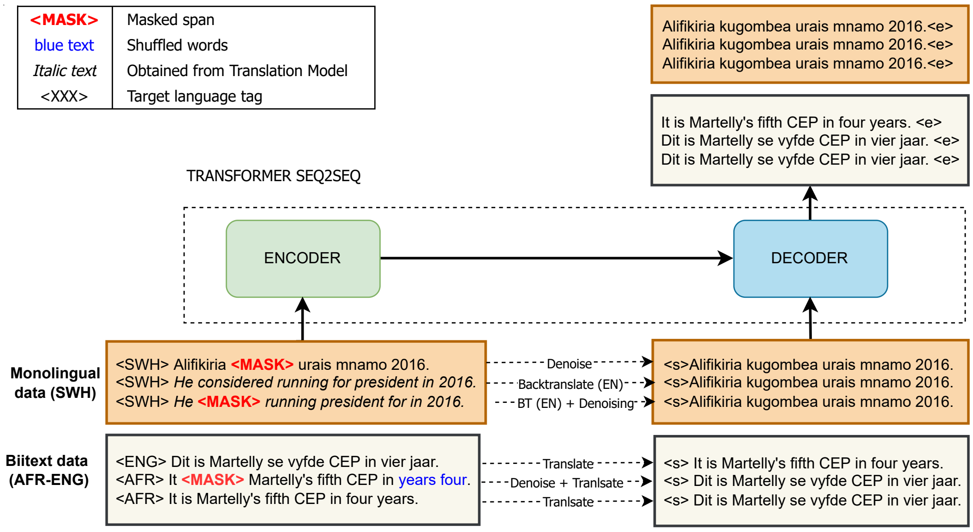

<MASK>
Masked span
blue text
Shuffled words
Italic text
Obtained from Translation Model
<XXX>
Target language tag
Alifikiria kugombea urais mnamo 2016.<e>
Alifikiria kugombea urais mnamo 2016.<e>
Alifikiria kugombea urais mnamo 2016.<e>
It is Martelly's fifth CEP in four years. <e>
Dit is Martelly se vyfde CEP in vier jaar. <e>
Dit is Martelly se vyfde CEP in vier jaar. <e>
TRANSFORMER SEQ2SEQ
ENCODER
DECODER
Monolingual data (SWH)
<SWH> Alifikiria <MASK> urais mnamo 2016.
<SWH> He considered running for president in 2016.
<SWH> He <MASK> running president for in 2016.
Denoise
Backtranslate (EN)
BT (EN) + Denoising
<s>Alifikiria kugombea urais mnamo 2016.
<s>Alifikiria kugombea urais mnamo 2016.
<s>Alifikiria kugombea urais mnamo 2016.
Biitext data (AFR-ENG)
<ENG> Dit is Martelly se vyfde CEP in vier jaar.
<AFR> It <MASK> Martelly's fifth CEP in years four.
<AFR> It is Martelly's fifth CEP in four years.
Translate
Denoise + Tranlsate
Tranlsate
<s> It is Martelly's fifth CEP in four years.
<s> Dit is Martelly se vyfde CEP in vier jaar.
<s> Dit is Martelly se vyfde CEP in vier jaar.

Figure 1: Pre-training Overview

In its multilingual pre-training strategy, mBART uses a denoising objective (Liu et al., 2020; Lewis et al., 2020) on combined monolingual corpora of several languages to train a Transformer sequence-to-sequence model (Vaswani et al., 2017). This strategy reduces the dependency on bitext by learning meaningful representations for multiple languages. The pre-trained model is fine-tuned for MT using bitext data. While this methodology does result in improved MT performance, the pre-training objective used does not induce the representation of similar sentences across languages to align, since it uses only monolingual data (Lin et al., 2020).

In contrast, mRASP2 combines the use of monolingual and bitext corpora. Their pre-training methodology is geared towards closing the representation gap across languages, bringing words and phrases with similar meanings across languages closer in the feature space. This results in better multilingual translation performance (Pan et al., 2021). To induce cross-language representations, mRASP2 uses word or phrase level dictionaries to augment both monolingual and bitext data, by replacing randomly chosen tokens in the source sentence by its corresponding words in another language. Since in this work we address primarily low-resource or under-represented languages, separate dictionaries are non-trivial to construct.

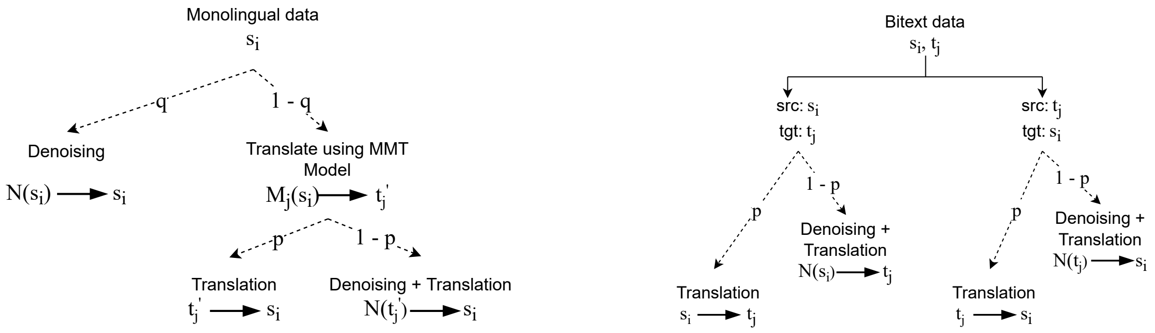
M2M-100 is a massively multilingual model trained on heuristically mined massive bitext corpora spanning 100 languages and 9,900 language pairs (Fan et al., 2021). Fine-tuning M2M-100 with bitext data from the shared task results in minimal improvement over a multilingual model trained from scratch (Section 6). We hypothesize that this is due to M2M-100's subword tokenizer. Since a majority of African languages are written in the Latin script, several subword units are common to many languages. For example, using the M2M-100 tokenizer in the WMT dataset, about 96% of the distinct subwords in African languages also appear in English. Since English corpora dominate the M2M-100 training dataset, the learnt representation of these common tokens are influenced majorly by English, limiting the contribution of African languages.

To address all of the above, we propose DENTRA, which uses a novel pre-training strategy and is trained exclusively on languages from the shared task using both monolingual and bitext data. Inspired by mRASP2 (Pan et al., 2021), our pre-training objective is also designed to explicitly reduce the representation gap between different languages. Figure 1 shows an overview of our pre-training technique.

To measure the effect of pre-training, we fine-tune the pre-trained model in one-to-many and many-to-one configurations of multilingual MT . In three out of four setups, average BLEU score of fine-tuned DENTRA exceeds that of fine-tuned M2M-100 by up to 1.56 points.

2 Definitions and Model Architecture

Task Description: The constrained translation track under WMT-2022 (Adelani et al., 2022b) consists of the following subtasks: English to 22 African languages (eng→{afs}), 22 African languages to English ({afs}→eng), French to 4


Monolingual data
s_i
q
1 - q
Denoising
$N(s_i) \longrightarrow s_i$
Translate using MMT Model
$M_j(s_i) \longrightarrow t_j'$
p
1 - p
Translation
$t_j' \longrightarrow s_i$
Denoising + Translation
$N(t_j') \longrightarrow s_i$
Bitext data
s_i, t_j
src: s_i
tgt: t_j
src: t_j
tgt: s_i
p
1 - p
Translation
$s_i \longrightarrow t_j$
Denoising + Translation
$N(s_i) \longrightarrow t_j$
p
1 - p
Translation
$t_j \longrightarrow s_i$
Denoising + Translation
$N(t_j) \longrightarrow s_i$

(a) Monolingual data

(b) Bitext data

Figure 2: Pre-training Data Preparation: Pre-training Objective for each example in the corpora are determined using the above trees. Solid lines indicate that both paths are executed. Dotted lines from a node indicate that one of its child nodes are selected at random, with the probability distribution along the edges.

African languages (fra→{afs}), 4 African languages to French ({afs}→fra) and 48 African to African languages within geographical and cultural clusters ({afs}→{afs}). In all tasks, training datasets are from the shared task while the validation and the test sets are from FLORES 200 (Goyal et al., 2022).

Multilingual Machine Translation (MMT): An MMT employs a sequence-to-sequence model to translate between arbitrarily many language pairs (Firat et al., 2016; Aharoni et al., 2019). We denote the set of languages in our corpora as $\mathcal{L} = \{l_1, l_2, \dots l_n\}$ and the bitext data as $\mathcal{D} = \{\mathcal{D}(l_i, l_j), \quad l_i, l_j \in \mathcal{L}\}$ where $\mathcal{D}(l_i, l_j) = \{(s_i, t_j)\}$ is the parallel corpus for languages l_i and l_j. Monolingual data is denoted $\mathcal{M} = \{\mathcal{M}(l_i), l_i \in \mathcal{L}\}$, and $s_i \in \mathcal{M}(l_i)$ denotes an example in the monolingual corpus for language l_i. For training MMT on bitext data $\mathcal{D}$, an artificial token indicating the target language is prefixed to the source, so that $(s_i, t_j) \in \mathcal{D}(l_i, l_j)$ becomes (<J>s_i, t_j) (Johnson et al., 2017). MMT can be trained in three configurations: one-to-many (**1**→**M**), many-to-one (**M**→**1**) and many-to-many (**M**→**M**) (Tang et al., 2021).

Model Architecture: We use the Transformer *big* architecture described in (Vaswani et al., 2017), with 6 encoder and decoder layers, 16 attention heads, and 1024 model dimension. We train our models using FAIRSEQ (Ott et al., 2019) toolkit, and other hyperparameter values listed in Appendix A.1.

3 Methodology

Our overall methodology employs the pre-training followed by fine-tuning pipeline used in prior work in NMT. (Liu et al., 2020; Lin et al., 2020). We present the pre-training strategy used in this work in Section 3.1 and discuss the fine-tuning configurations used in our submission in Section 3.2.

3.1 Pre-training

In our pre-training, we combine monolingual and bitext data in the same corpus. The objective of pre-training is to either denoise, or translate, or both. For each individual example in the corpus, we randomly select which of these objectives to apply. By interleaving denoising and translation, our goal is to drive the model towards learning cross-lingual representations while at the same time learning robust semantic representations. The strong cross lingual representations enable better few-shot and zero-shot translation performance (Pan et al., 2021). Figure 2 illustrates our pre-training methods for both monolingual and bitext data. $N(\cdot)$ is the noising function which we describe in detail in Section 3.1.3, while $M_j(\cdot)$ denotes the translation function using an MMT model for translation to language l_j. In the remainder of this section, we describe each component of the pre-training individually.

3.1.1 Monolingual Data

Figure 2a shows the two ways in which we utilize monolingual data in pre-training. Independently, for each monolingual example $s_i \in \mathcal{M}(l_i)$ one of denoising or translation is selected with probability q and $1 - q$ respectively.

If denoising is selected, we apply a denoising objective similar to mBART (Liu et al., 2020) by masking or shuffling randomly chosen spans. If translation is selected, s_i is first translated to all languages l_j for which $\mathcal{D}(l_i, l_j) \in \mathcal{D}$. The translation is done by MMT models which are obtained by training Transformer models from scratch on $\mathcal{D}$, described in Section 5.1. Lets $M_j(s_i) = t'_j$ is the translation of s_i into language l_j. For the pair (s_i, t'_j), either translation only, or denoising + translation is selected with probabilities p and $1-p$ respectively. If translation is chosen, s_i must be reconstructed from t'_j, following traditional back-translation (Sennrich et al., 2016). If denoising + translation is selected, then s_i must be reconstructed from the noised version $N(t'_j)$. In this manner, the pre-training also incorporates back-translation for utilizing monolingual data.

3.1.2 Bitext Data

Figure 2b shows the bitext data usage in pre-training. Given a pair of sentences (s_i, t_j), the pre-training procedure treats s_i as source and t_j as target, and vice versa. Therefore, two pre-training examples are generated for each example in the bitext data. This is in contrast to pre-training with monolingual data described above, where only one pre-training example was generated per input example. Having designated either s_i or t_j as source, the pre-training procedure follows a path similar to that of monolingual pre-training with backtranslation.

3.1.3 Noising Function

The noising function $N(\cdot)$ largely follows the noising techniques used in Liu et al. (2020). Given an input sentence s, $N(s)$ randomly selects a noising type and applies it on s.

Mask only: With probability p_m, $N(s)$ applies span masking on s. It randomly samples spans of tokens from s, with length of the span drawn from a geometric distribution with parameter m_{sl}, and clipped at 3. The fraction of tokens thus masked is at most m_{sr}. Each masked span is either replaced by a single <MASK> token, or deleted, or replaced by randomly selected word in another language with equal probabilities.

Shuffle only: With probability p_s, $N(s)$ applies shuffling to s. An s_r fraction of tokens are selected at random from tokens in s, and permuted among each other, leaving the unsampled tokens intact.

Mask and Shuffle: With probability p_{ms}, $N(s)$ applies both masking and shuffling to s. First, masking is applied as described above. During the shuffling step, <MASK> tokens are excluded from the tokens to be sampled for shuffling.

None: With probability $1 - p_m - p_s - p_{ms}$, no noising is applied on the input.

3.1.4 Combining Datasets

We combine both $\mathcal{D}$ and $\mathcal{M}$ in pre-training. In order to balance the training dataset across language pairs, we apply temperature based sampling following (Fan et al., 2021) with one major change. Since we are operating in a data constrained setting, we do not reduce the size of any dataset.

Let $N_{(i,j)} = |\mathcal{D}(l_i, l_j)|$ the size of the bitext $\mathcal{D}(l_i, l_j)$, and $N_{\mathcal{D}} = \sum_{(i,j)} N_{(i,j)}$. Then the scaled proportion of language pair (l_i, l_j) is $\alpha'_{i,j} = \frac{\alpha_{(i,j)}}{\sum_{(i,j)} \alpha_{(i,j)}}$ where $\alpha_{(i,j)} = \left(\frac{N_{(i,j)}}{N_{\mathcal{D}}}\right)^{\alpha}$. The rescaled size of language pair (l_i, l_j) is then $R_{(i,j)} = max(N_{(i,j)}, \alpha'_{(i,j)} N_{(i,j)})$

We train the transformer network on the combined dataset until convergence, and then select the best checkpoints for further fine-tuning.

3.2 Fine-tuning

For fine-tuning our pre-trained models, we use bitext data $\mathcal{D}$. Unlike pre-training, we don't noise the source side at all. We apply fine-tuning in 1→M and M→1 settings. Following the pre-training setup, we continue to prefix the tag for the target language in the source side, and also rebalance the datasets as in Section 3.1.4. The checkpoint with best BLEU score on validation set is used for final translations. After fine-tuning, we have the following models. (i) eng→{afs}, (ii) {afs}→eng, (iii) fra→{afs}, and (iv) {afs}→fra. For the remaining pairs, i.e. between African languages, we use DENTRA directly.

4 Datasets and Pre-processing

For all experiments, we employ the datasets provided by the organizers, which mainly consist of datasets from Opus (Tiedemann, 2012), Mafand (Adelani et al., 2022a) and Web crawled aligned through LASER (Heffernan et al., 2022). It is worth mentioning that, in all pre-training and fine-tuning we only used a monolingual corpus of 26 languages and English and French-centric bitext. We have not used any African to African bitext in our experiments. Prior to using for pre-training or fine-tuning, datasets were filtered, cleaned, and preprocessed.

4.1 Data Filtering

Based on characteristics of the dataset and a few observed issues, we employed several heuristics to reject highly noisy examples from $\mathcal{D}$ and $\mathcal{M}$. Given an example $(s_i, t_j) \in \mathcal{D}(l_i, l_j)$, we reject it if (i) $|s_i| < 3$ or $|t_j| < 3$, (ii) $|s_i| > 1000$ or $|t_j| > 1000$, (iii) a character other than . appears at least 5 consecutive times in either s_i or t_j, (iv) a word other than . appears at least 3 consecutive times in either s_i or t_j, (v) s_i is identical to t_j, (vi) $|s_i|/|t_j| < 0.2$ or > 5, (vii) langid of s_i or t_j is not the expected langid with a confidence of at least 80% (where langid is computed using fasttext[2]), (viii) the fraction of characters not belonging in this language are more than 50% [3]. For monolinugal data $\mathcal{M}$, we apply rules corresponding to (i)-(iv) , and (vii)-(viii) above.

After filtering and pre-processing the size of the datasets (combined by English centric, French centric, or monolingual) obtained are shown in Table 1[4]. Full list is shown in Appendix A.2.

Dataset	# Datasets	Total Size	Min	Max	Δ
eng-{afs}	22	109.36	0.21	32.01	21.32 %
fra-{afs}	4	13.40	0.22	11.51	3.96 %
Mono	26	34.17	0.0	12.73	0.58 %

Table 1: Data set sizes specified in Million sentence pairs (or sentences). Δ refers to the percentage of sentence pairs (or sentences) rejected after filtering and pre-processing

Param	p	q	p_m, p_s, p_{ms}	m_{sl}	m_{sr}	s_r	α
Value	0.25	0.66	0.25	0.15	0.2	0.05	0.7

Table 2: Hyper-parameter values used in our data preparation

5 Experimental Setup

Table 2 specifies the hyperparameters we have used in pre-training (Section 3.1). We train the model for 6 epochs and select the best checkpoint based on pre-training task performance on a held out validation set.

The best pre-training checkpoint was subsequently fine-tuned for various tasks where we consider the concatenation of all FLORES 200 *dev sets* for the relevant translation directions as validation set. The FLORES *dev sets* have 997 examples for all language pairs.

For evaluation, we use two test sets. The FLORES *devtest* set (subsequently, we refer to it as FLORES *test set*), which has 1012 examples for all language pairs, and an in-domain test set, which is randomly sampled from the provided bitext data and has about 5000 examples from each language pair. Unless otherwise specified, we report performances on the FLORES *test set*.

We use tokenized BLEU from Moses[5] to measure the performance of all translations. Prior to computing BLEU we word tokenize all translations, also using Moses.

5.1 Models

We prepared following baselines for the comparison of our pre-trained and fine-tuned models:

MMT[6] is trained from scratch separately for four tasks with their corresponding bitext: eng→{afs}, {afs}→eng, fra→{afs} and {afs}→fra. We evaluate it in 1→M and M→1 setups.

M2M-100 is trained on many-to-many datasets of 100 languages. We use the trained version provided by the authors (Fan et al., 2021) and evaluate it in all setups, 1→M, M→1, and M→M. Note that M2M-100 does not support all language pairs in this task and thus, we report performance on only the common language pairs. In particular, M2M-100 includes 14/22 languages in eng↔{afs}, 3/4 in fra↔{afs} and 22/48 in {afs}→{afs}. In all M2M-100 experiments, we employ its 418M parameters checkpoint.

M2M_FT employs the pre-trained checkpoint of M2M-100 and fine-tunes it with bitext data. Similar to Adelani et al. (2022a), unseen African languages {*kam, kin, luo, nya, orm, sna, tso, umb*} are mapped to {*km , ht, lo, yi, fy, ba, kk, uz*} respectively for fine-tuning M2M-100. M2M_FT is used for evaluations in 1→M and M→1 setups only.

The following are the models trained in this work for demonstrating the importance of our pre-training and fine-tuning strategies.

DENTRA is the pre-trained model described in Section 3, which we train using (i) monolingual data in 26 languages and (ii) bitext data for only English and French centric directions. We compare DENTRA against the corresponding baselines in all setups, 1→M, M→1, and M→M.

DENTRA_FT uses bitext data for English and French centric directions to fine-tune the DENTRA

[2] https://fasttext.cc/docs/en/python-module.html

[3] Character sets for each language are built by manually curating distinct characters obtained from $\mathcal{D}$

[4] For the Kinyarwanda language, no monolingual data was provided. We reused the Kinyarwanda side from the Kinyarwanda-English bitext for this purpose. For English and French, we randomly sampled 1 million sentences from the combined English/French sides of the bitext datasets provided.

[5] https://github.com/moses-smt/mosesdecoder

[6] These models are also used for backtranslation described in Section 3.1.1

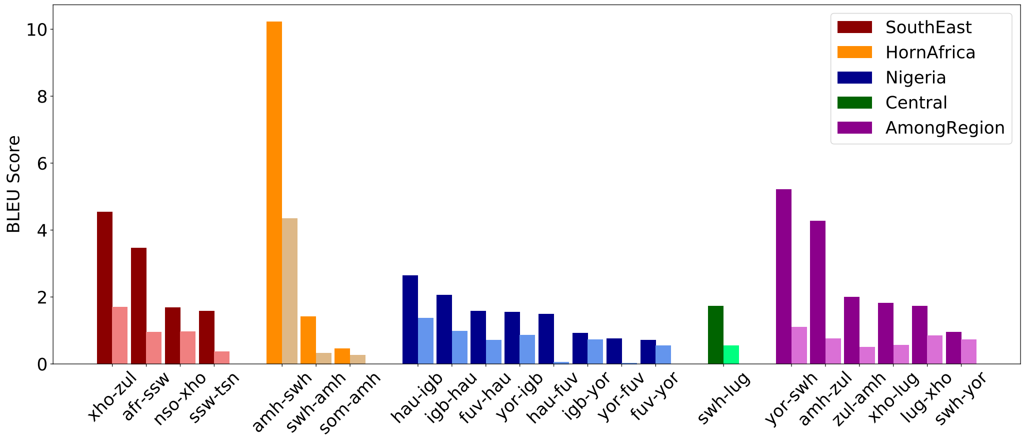


Figure 3: BLEU scores for M2M-100 and DenTra (zero-shot) among African languages on the Flores 200 test set. Dark colors represent DenTra and light colors represent M2M-100.

Tasks	M2M-100	DenTra
eng→{afs}	4.51	9.13
fra→{afs}	4.74	9.48
{afs}→eng	9.13	22.63
{afs}→fra	8.21	15.28

Table 3: Average performance for common languages of DenTra and M2M-100 before fine-tuning

model. DenTra_FT is trained and evaluated against the baselines in only the 1→M and M→1 setups.

6 Comparisons with Baselines

6.1 Without Fine-tuning

In this section, we will show the advantage of DenTra over M2M-100 for the 26 languages in the task without any fine-tuning on either models. As DenTra and M2M-100 both include bitext in their training, we can directly use them for translation.

Table 3 shows the average performance of M2M-100 and DenTra for the 14 English and 3 French centric tasks in M→1 and 1→M setups. In all four tasks, DenTra outperforms M2M-100 by significant margins.

Furthermore, in Figure 3 we display the performance of M2M-100 and DenTra on {afs}→{afs} tasks, i.e. translation between African languages. Similar to M→1 and 1→M setup, we include only the 22 common directions of DenTra and M2M-100 (Full performance list for DenTra is provided in Appendix A.3). Note this is the *zero-shot setting* (Johnson et al., 2017) for both[7]. However, in all translation directions, DenTra outperforms M2M-100 by large margins. These results shows the advantage of pre-training with combined monolingual and bitext data for only the desired set of languages, over pre-training with a large number of additional languages. This confirms our hypothesis discussed in Section 1.

6.2 With Fine-tuning

We evaluate DenTra after it is fine-tuned on the four M→1 and 1→M tasks. Tables 4, 5, 6, and 7 show the BLEU scores for DenTra_FT along with all baselines. Following conclusions may be drawn:

All model variants including DenTra_FT, M2M_FT and DenTra are significantly better than M2M-100 across all language pairs. The general trend of performance comparison in best to worst order is DenTra_FT, M2M_FT, MMT, DenTra, M2M-100, except {afs}→fra where M2M_FT outperforms DenTra_FT.

Improvement of M2M_FT and DenTra_FT over MMT shows the advantage of fine-tuning after pre-training in general. Moreover, since DenTra_FT typically outperforms M2M_FT also, this demonstrates the advantage of including the monolingual corpora and denoising objectives in the pre-training phase.

Generally, it has been shown that combining low resource and high resource languages in a single translation model benefits the low resource languages. We also observe similar behavior as shown by our MMT model. However, we find that extreme multilingual models like M2M-100 must necessar-

[7] Our assumption is that M2M-100 is zero shot in these settings.

Model	$xxx \rightarrow$ **eng**											
	afr	amh	ful	hau	ibo	kam	kin	lug	luo	nso	nya	orm
MMT	**54.59**	22.47	5.51	25.85	19.67	8.38	22.9	15.77	16.58	30.78	21.21	10.61
M2M-100	43.48	6.64	1.98	6.4	5.62	-	-	2.63	-	4.05	-	-
M2M_FT	53.42	22.87	5.62	23.44	18.34	6.65	20.96	14.17	14.79	29.3	20.56	9.88
DenTra	51.65	18.85	5.36	23.24	16.73	8.59	22.24	15.12	14.81	27.85	19.57	9.34
DenTra_FT	54.46	**23.7**	**5.78**	**26.64**	**20.05**	**9.26**	**22.85**	**16.25**	**16.89**	**31.53**	**21.8**	**10.64**
	sna	som	ssw	swh	tsn	tso	umb	xho	yor	zul	**AVG**	**MED**
MMT	21.7	19.6	23.36	37.35	21.21	23.51	5.27	30.38	**13.85**	31.12	21.89	21.46
M2M-100	-	2.94	4.95	25.62	0.78	-	-	10.35	1.93	10.4	9.13	5.28
M2M_FT	21.29	18.1	23.33	36.54	20.47	22.13	4.95	29.59	11.46	29.93	20.81	20.76
DenTra	19.77	16.77	20.95	34.16	18.5	21.72	4.9	27.86	11.65	28.08	19.9	19.21
DenTra_FT	**22.53**	**19.66**	**23.73**	**38.77**	**21.47**	**23.71**	**5.32**	**31.01**	13.72	**32.41**	**22.37**	**22.16**

Table 4: BLEU score on the Flores 200 test set, before and after fine-tuning for English centric MT. For each subtask, the **best model** is bold

Model	**eng** $\rightarrow xxx$											
	afr	amh	ful	hau	ibo	kam	kin	lug	luo	nso	nya	orm
MMT	40.65	**9.12**	0.66	**22.64**	15.41	**3**	14.52	**6.04**	6.49	23.71	13.82	**2.1**
M2M-100	26.99	0.51	0.25	2.46	2.53	-	-	1.09	-	0.65	-	-
M2M_FT	38.62	6.85	0.64	18.58	11.72	2.93	14.22	5.93	6.94	**24.65**	12.38	1.91
DenTra	40.38	4.45	**0.76**	21.53	13.72	2.43	12.79	5.53	6.9	21.97	13.13	1.72
DenTra_FT	**40.95**	8.42	0.64	22.6	**15.69**	2.68	**14.54**	5.97	7.29	24.55	**14.14**	2.06
	sna	som	ssw	swh	tsn	tso	umb	xho	yor	zul	**AVG**	**MED**
MMT	11.57	**9.83**	**7.48**	33.92	18.43	**16.3**	0.96	**15.76**	**2.52**	15.08	13.18	12.7
M2M-100	-	0.49	1.05	19.35	2.61	-	-	2.03	1.07	2.12	4.51	1.56
M2M_FT	10.33	9.15	7.21	29.43	17.17	16.24	1.15	14.23	2.2	12.96	12.07	11.03
DenTra	10.73	8.13	6.05	33.08	16.23	13.08	1	13.61	2.39	13.92	11.98	11.76
DenTra_FT	**11.68**	9.79	7.42	**34.69**	**18.45**	16.24	**1.16**	**15.76**	2.41	**15.32**	**13.29**	**12.91**

Table 5: BLEU score on the Flores 200 test set, before and after fine-tuning for English centric MT. For each subtask, the **best model** is bold

Model	$xxx \rightarrow$ **fra**					
	lin	kin	swh	wol	**AVG**	**MED**
MMT	15.46	17.61	27.29	9.69	17.51	16.54
M2M-100	2.78	-	19.79	2.07	8.21	2.78
M2M_FT	**17.02**	**18.4**	28.22	**11.44**	**18.77**	**17.71**
DenTra	14.38	16.13	22.14	9.33	15.5	15.25
DenTra_FT	16.27	18.28	**28.64**	11.28	18.62	17.27

Table 6: BLEU score on the Flores 200 test set, before and after fine-tuning for French centric MT. For each subtask, the **best model** is bold

Model	**fra** $\rightarrow xxx$					
	lin	kin	swh	wol	**AVG**	**MED**
MMT	13.4	10.08	21.45	4.56	12.37	11.74
M2M-100	0.93	-	12.88	0.42	4.74	0.93
M2M_FT	14.32	10.67	21.1	4.54	12.66	12.49
DenTra	4.97	9.76	21.02	2.46	9.55	7.365
DenTra_FT	**14.4**	**10.85**	**22.09**	**5.09**	**13.11**	**12.62**

Table 7: BLEU score on the Flores 200 test set, before and after fine-tuning for French centric MT. For each subtask, the **best model** is bold

ily have larger capacity to represent all languages in its corpora. In particular, if the languages of interest are restricted, it is better to also restrict pre-training to these languages only (Adelani et al., 2022a).

Finally, we note that the performance of translation models where African languages are the target language are generally lower than those where English or French are the target language. This is expected, since the volume of data where each individual African language appears on the target side is much lower than English or French.

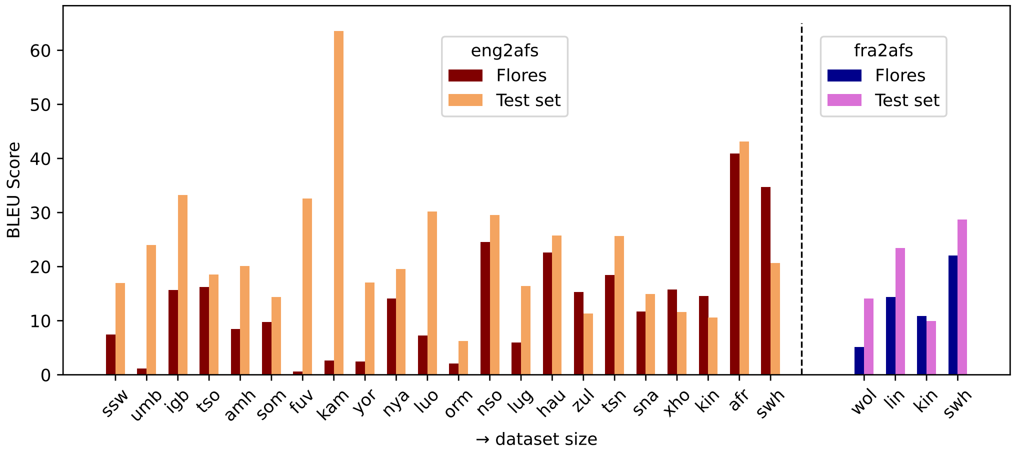


Figure 4: BLEU score comparison of the DENTRA_FT model for FLORES 200 and in-domain test set (isolated from bitext training data $\mathcal{D}$ for English/ French to African languages.

7 Analysis

In this section, we conduct a set of analytical experiments to better understand the datasets and what contributes to performance gains.

Figure 4 shows the BLEU scores of the DenTra_FT model on {eng,fra}→{afs} translation directions, on the both FLORES 200 test set and the in-domain test set sampled from the bitext data prior to training. The language pairs on the horizontal axis are ordered by the dataset size (left to right in increasing order) independently for English and French centric directions. For the English centric translation (eng→{afs}), the BLEU scores on both test sets have little correlation with the dataset size, indicating noisy data. Some languages, such as *umb, fuv, kam,* and *yor* have stark difference between FLORES and in-domain test sets, indicating that these datasets may have predictable patterns that have no relevance to the translation of these languages. This is further exemplified by the comparison to *tso*, which has a smaller dataset yet exhibits better generalization.

Further investigations reveal two primary problems with the bitext data. First, some of these languages have several duplicates in the African side of the data. For example, for Kamba-English (*kam-eng*) dataset, the distinct number of Kamba sentences is less than 5% of the total dataset size. However, this is not consistent across all languages exhibiting overfitting on the training data, as the number of distinct Yoruba (*yor*) sentences in its bitext is about 95% of the total dataset.

Second, the African side of many datasets contain a large fraction of Indic languages from the Social Media domain. Strict heuristics designed based on manual inspection by the authors rejected about $637,000$ examples as being clearly in the Hindi language. Clearly, neither langid nor the LASER encoder (Schwenk and Douze, 2017) are able to reliably detect and align data for these languages. We postulate that low resource languages form a vicious cycle for MT systems trained on bitext data created using multilingual encoders. This opens avenues for future work to explore bitext creation for low resource languages.

8 Conclusion

DenTra has shown significant performance gains in Multilingual Machine Translation for African languages as demonstrated in this paper. DenTra integrates denoising, backtranslation, and translation into the same pre-training setup, and has helped to improve MT performance after fine-tuning for both English and French centric translation. We have shown that massively multilingual models like M2M-100 may not be a good choice for fine-tuning when the languages to be translated from/to are restricted to a small set. Finally, we have studied the variation in performance and reported issues seen in heuristically created bitext data. While this is a known issue, we show this problem to be exacerbated for low-resource languages that share the alphabet with high-resource ones.

References

David Adelani, Jesujoba Alabi, Angela Fan, Julia Kreutzer, Xiaoyu Shen, Machel Reid, Dana Ruiter, Dietrich Klakow, Peter Nabende, Ernie Chang, Tajuddeen Gwadabe, Freshia Sackey, Bonaventure F. P. Dossou, Chris Emezue, Colin Leong, Michael Beukman, Shamsuddeen Muhammad, Guyo Jarso, Oreen Yousuf, Andre Niyongabo Rubungo, Gilles Hacheme, Eric Peter Wairagala, Muhammad Umair Nasir, Benjamin Ajibade, Tunde Ajayi, Yvonne Gitau, Jade Abbott, Mohamed Ahmed, Millicent Ochieng, Anuoluwapo Aremu, Perez Ogayo, Jonathan Mukiibi, Fatoumata Ouoba Kabore, Godson Kalipe, Derguene Mbaye, Allahsera Auguste Tapo, Victoire Memdjokam Koagne, Edwin Munkoh-Buabeng, Valencia Wagner, Idris Abdulmumin, Ayodele Awokoya, Happy Buzaaba, Blessing Sibanda, Andiswa Bukula, and Sam Manthalu. 2022a. A few thousand translations go a long way! leveraging pre-trained models for African news translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

David Ifeoluwa Adelani, Md Mahfuz Ibn Alam, Antonios Anastasopoulos, Akshita Bhagia, Marta Costa-Jussá, Jesse Dodge, Fahim Faisal, Christian Federmann, Natalia Fedorova, Francisco Guzmán, Sergey Koshelev, Jean Maillard, Vukosi Marivate, Jonathan Mbuya, Safiyyah Saleem, and Holger Schwenk. 2022b. Findings of the WMT 2022 shared task on large-scale machine translation evaluation for african languages. In *Proceedings of the Seventh Conference on Machine Translation*.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig. 2020. Ethnologue: Languages of the world. twenty-third edition. In *eds*.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Chris Chinenye Emezue and Bonaventure F. P. Dossou. 2021. MMTAfrica: Multilingual machine translation for African languages. In *Proceedings of the Conference on Machine Translation*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*.

Kevin Heffernan, Onur Çelebi, and Holger Schwenk. 2022. Bitext mining using distilled sentence representations for low-resource languages. *arXiv preprint arXiv:2205.12654*.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. 2020. Pre-training multilingual neural machine translation by leveraging alignment information. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the Workshop on Representation Learning for NLP*.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the*

Annual Meeting of the Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems*.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

A Appendix

A.1 Model Hyper-parameters

For pre-training and fine-tuning the DENTRA, we base our experiments on FAIRSEQ toolkit. Table 8 presents the hyper-parameter values used in all experiments. For fine-tuning, we employ the best checkpoint obtained from pre-training and continue to train them without resetting the *lr-scheduler*.

A.2 Languages in the Dataset

Table 9 shows the languages used in our experiments along with their bitext and monolingual data sizes.

A.3 Cluster wise Performance

Table 10 shows the performance of DENTRA and M2M-100 on the FLORES 200 test set for different African language pairs clustered geographically/culturally.

Params	Values
optimizer	adam
adam-betas	’(0.9, 0.98)’
clip-norm	0.0
lr	0.0005
lr-scheduler	inverse_sqrt
warmup-updates	4000
warmup-init-lr	1e-07
dropout	0.3
criterion	label_smoothed_cross_entropy
label-smoothing	0.1
max-tokens (batch size)	3584
num-updates (Pre-training)	1609115
num-updates (Fine-tuning) eng-{af}	548227
num-updates (Fine-tuning) fra-{af}	49600

Table 8: Model hyper-parameters and their values

Languages	ISO	Bitext Size	Monolingual Size	Rejected Bitext Size
Afrikaans	afr	13.9	12.732	0.18
Amharic	amh	1.02	0.006	0.11
Nigerian Fulfulde	fuv	1.3	0.255	0.15
Hausa	hau	3.6	3.513	5.5
Igbo	ibo	0.4	0.452	0.1
Kamba	kam	1.58	0.01	0.08
Kinyarwanda	kin	9.6 (eng)	-	0.36 (eng)
		1.2 (fra)	-	0.15 (fra)
Luganda	lug	3.39	0.11	0.11
Luo	luo	2.6	0.035	0.16
Northern Sotho	nso	2.9	0.018	0.18
Chichewa	nya	1.7	0.261	0.14
Oromo	orm	2.7	0.134	0.13
Swati	ssw	8.6	0.257	0.02
Shona	sna	1.25	0.007	0.3
Somali	som	0.2	-	0.15
Swahili	swh	31.7 (eng)	12.642	0.8 (eng)
		11.4 (fra)	-	0.3 (fra)
Tswana	tsn	5.6	0.04	0.43
Xitsonga	tso	0.6	0.037	0.05
Umbundu	umb	0.2	0.043	0.1
Xhosa	xho	9.3	0.308	19.78
Yoruba	yor	1.6	0.51	0.1
Zulu	zul	3.9	0.557	0.23
Lingala	lin	0.3	0.042	0.06
Wolof	wol	0.2	0.206	0.03
English	eng	-	1	0
French	fra	-	1	0

Table 9: Languages, their ISO codes used in the paper, and their corresponding data sizes (in Million sentences)

.

Cluster ID	Task	DENTRA	M2M-100
A	xho→zul	4.54	1.7
	zul→sna	2.71	-
	sna→afr	10.65	-
	afr→ssw	3.47	0.95
	ssw→tsn	1.58	0.37
	tsn→tso	2.28	-
	tso→nso	3.8	-
	nso→xho	1.69	0.96
B	swh→am	1.41	0.33
	amh→swh	10.22	4.35
	luo→orm	0.63	-
	som→amh	0.46	0.26
	orm→som	0.85	-
	swh→luo	2.4	-
	amh→luo	3.51	-
	luo→som	2.37	-
C	hau→ibo	2.64	1.37
	ibo→yor	0.92	0.72
	yor→fuv	0.76	0.03
	fuv→hau	1.58	0.71
	ibo→hau	2.06	0.98
	yor→ibo	1.55	0.86
	fuv→yor	0.71	0.55
	hau→fuv	1.49	0.06
	wol→hau	2.11	-
	hau→wol	1.57	-
	fuv→wol	1.24	-
	wol→fuv	1.22	-
D	kin→swh	3.42	-
	lug→lin	1.87	-
	nya→kin	2.22	-
	swh→lug	1.73	0.55
	lin→nya	2.44	-
	lin→kin	2.45	-
	kin→lug	1.96	-
	nya→swh	3.08	-
E	amh→zul	4.27	0.75
	yor→swh	5.21	1.1
	swh→yor	0.95	0.72
	zul→amh	2	0.5
	kin→hau	2.53	-
	hau→kin	2.06	-
	nya→som	3.27	-
	smo→nya	4.28	-
	xho→lug	1.82	0.56
	lug→xho	1.73	0.85
	wol→swh	3.76	-
	swh→wol	1.71	-

Table 10: BLEU scores on the FLORES 200 test set for geographical/cultural clusters. A: South/South East Africa, B: Horn of Africa, C: Nigerian, D: Central African, E: Among the regions

The VolcTrans System for WMT22 Multilingual Machine Translation Task

Xian Qian[1], **Kai Hu**[1], **Jiaqiang Wang**[1], **Yifeng Liu**[2]
Xingyuan Pan[3], **Jun Cao**[1], **Mingxuan Wang**[1]
[1] ByteDance AI Lab, [2] Tsinghua University, [3] Wuhan University
{qian.xian, hukai.joseph,wangjiaqiang.sonian,
caojun.sh, wangmingxuan.89}@bytedance.com
liuyifen20@mails.tsinghua.edu.cn, panxingyuan209@gmail.com

Abstract

This report describes our VolcTrans system for the WMT22 shared task on large-scale multilingual machine translation. We participated in the unconstrained track which allows the use of external resources. Our system is a transformer-based multilingual model trained on data from multiple sources including the public training set from the data track, NLLB data provided by Meta AI, self-collected parallel corpora, and pseudo bitext from back-translation. A series of heuristic rules clean both bilingual and monolingual texts. On the official test set, our system achieves 17.3 BLEU, 21.9 spBLEU, and 41.9 chrF2++ on average over all language pairs. The average inference speed is 11.5 sentences per second using a single Nvidia Tesla V100 GPU. Our code and trained models are available at https://github.com/xian8/wmt22

1 Introduction

Multilingual Machine Translation attracts much attention in recent years due to its advantages in sharing cross-lingual knowledge for low-resource languages. It also dramatically reduces training and serving costs. Training a multilingual model is much faster and simpler than training many bilingual ones. Serving multiple low-traffic languages using one model could drastically improve GPU utilization.

The WMT22 shared task on large-scale multilingual machine translation includes 24 African languages (Adelani et al., 2022b). Inspired by previous research works, we train a deep transformer model to translate all languages since large models have been demonstrated effective for multilingual translation (Fan et al., 2021; Kong et al., 2021; Zhang et al., 2020). We participated in the unconstrained track that allows the use of external data. Besides the official dataset for the constrained track, and the NLLB corpus provided by MetaAI (NLLB Team et al., 2022), we also collect parallel and monolingual texts from public websites and sources. These raw data are cleaned by a series of commonly used heuristic rules, and a minimum description length (MDL) based approach to remove samples with repeat patterns. Monolingual texts are used for back translation. For some very low-resource languages such as Wolof, iterative back-translation is adopted for higher accuracy.

We compare different training strategies to balance efficiency and quality, such as streaming data shuffling, and dynamic vocabulary for new languages. Furthermore, we used the open-sourced LightSeq toolkit [1] to accelerate training and inference.

On the official test set, our system achieves 17.3 BLEU, 21.9 spBLEU, and 41.9 chrF2++ on average over all language pairs. Averaged inference speed is 11.5 sentences per second using a single Nvidia Tesla V100 GPU.

2 Data

2.1 Data Collection

Our training data are mainly from four sources: the official set for constrained track, NLLB data provided by Meta AI, self-collected corpora, and pseudo training set from back translation.

For each source, we collect both parallel sentence pairs and monolingual sentences. A parallel sentence pair is collected if one side is in African language and the other is in English or French. We did not collect African-African sentence pairs as we use English as the pivot language for African-to-African translation. Instead, they are added to the monolingual set. More specifically, we split every sentence pair into two sentences and add them to the monolingual set accordingly. For example, the source side of a fuv-fon sentence pair is added to the fuv set. This greatly enriches the monolingual dataset, especially for the very low-resource

[1] https://github.com/bytedance/lightseq

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1068 - 1075
December 7-8, 2022 ©2022 Association for Computational Linguistics

languages.

We merge multiple corpora from the same source into one and use bloom filter [2](Bloom, 1970) for fast deduplication. To reduce false positive errors which over delete distinct samples, we set the error rate $1e-7$ and capacity of $4B$ samples which costs $100G$ host memory.

The official set includes the data from data track participants, OPUS collections, and the NLLB parallel corpora mined from Common Crawl (com) and other sources. All domains in OPUS collections are involved, such as Mozilla-I10n, which could introduce many noises such as programming languages, and needs extra rules to clean.

NLLB data provided by Meta AI has three subsets: primary bitext including a seed set that is carefully annotated for representative languages and a public bitext set downloaded from open sources and mined bitexts that are automatically discovered by LASER3 encoder in a global mining pipeline, back-translated data from a pretrained model. We add the first two subsets in our training set.

Some public bitext data that are no longer available or require authorization such as JW300 (Agić and Vulić, 2019), Lorelei[3] and Chichewa News [4] are not included. We noticed that the NLLB team released another version of mined data recently in hugging-face [5], which is different from the version on the WMT22 website. We merge the new version into the old one and remove duplicates.

We collected additional bitexts in two ways: large-scale mining from general web pages, and manually crawling from specific websites and sources.

Large-scale mining focused on two scenarios, parallel sentences appearing on a single web page such as dictionary web pages that use multiple bilingual sentences to exemplify the usage of a word, and parallel web pages that describe the same content but are written in different languages. We extract these pages from the Common Crawl corpus. Then we utilized Vecalign (Thompson and Koehn, 2019), an accurate and efficient sentence alignment algorithm to mine parallel bilingual sentences. We use LASER (Schwenk and Douze, 2017) encoders released by WMT to obtain multilingual sentence embeddings and facilitate the alignment work. We collected about 3 million sentence pairs namely LAVA corpus and submitted them to the data track. And another $150M$ pairs for the unconstrained track.

Specific websites and sources have fewer but higher-quality sentence pairs. For example, the bible website[6] labels the order of sentences across languages so we can align them easily without sentence segmentation. Since JW300 is not publicly available, we crawled pages from Jehovah's Witnesses[7] to recover the dataset.

Monolingual texts have richer sources such as VOA news in Amharic [8] and OSCAR (Abadji et al., 2022), which improve English/French → African translation using back-translation. Monolingual texts from parallel data are also collected as described above. For African → English/French translation, we clean Wikipedia pages in English/French to get monolingual texts. For languages that gain significantly from back-translation such as Wolof, we run another round of back-translation to generate high-quality pseudo data.

2.2 Data Cleaning

We used the following rules to clean parallel datasets, except the NLLB mined bitext.

- Filter out parentheses and texts in between if the numbers of parentheses in two sentences are different.
- Filter out sentence pairs if numbers mismatch or one sentence ends with punctuation *: ! ? ...* and the other mismatches.
- Filter out sentences shorter than 30 characters, sentences having URLs or emails, or words longer than 100 characters.
- De-duplication: remove sentence pairs sharing the same source or target but having different translations.
- Sentences having programming languages are removed. We manually create a set of keywords to detect programming languages, such as *if (* , *==* and *.getAttribute* .
- Language identification using the NLLB language identification model trained by fastText (Joulin et al., 2017)

[2] https://pypi.org/project/bloom-filter

[3] https://catalog.ldc.upenn.edu/LDC2021T02

[4] https://zenodo.org/record/4315018#.YypJWezML0p

[5] https://huggingface.co/datasets/allenai/nllb

[6] https://www.bible.com/languages

[7] https://www.jw.org

[8] https://amharic.voanews.com/

One type of noisy text could survive the rules above, which has repeat patterns and commonly exists in many datasets. Here are some examples,

Download Bongeziwe Mabandla mini esadibana ngayo (#001) Mp3 Bongeziwe Mabandla - mini esadibana ngayo (#001).

Coaster Gift,Paper-Cut Coaster Zodiac,Red Coaster Cute,Paper-Cut Zodiac Coaster

mm mm mm MPEE(um) MPEP(um) mm mm mm mm mm mm kg kg

A natural choice to detect these repeating patterns is the minimum description length (MDL) which finds the optimal compression by encoding frequent substrings with shorter codes.

Specifically, given a sentence $\mathbf{s}$, our MDL objective minimizes the length of the codebook plus the bits to encode the sentence:

$$\mathrm{MDL}(\mathbf{s}) = \min_{\mathbf{s}=w_1 w_2 \ldots w_n} \left(C \sum_{\text{distinct } w} |w| - \sum_i \log \left(p(w_i | w_{i-1}) \right) \right)$$

where $w_1, w_2, \ldots w_n$ is the word (coding entry) sequence, C is a positive constant, which balances the contribution of the codebook and length of the encoded sequence. $|w|$ is the length of word w. In our experiments, we set $C = 2$. $p(w_i|w_{i-1}) = \frac{\#w_{i-1}w_i}{\#w_{i-1}}$ is the conditional probability of word bigrams in the sequence.

A sentence is noisy if the ratio of MDL over sentence length is less than a predefined threshold:

$$\mathbf{s} \text{ is noisy if } \quad \frac{\mathrm{MDL}(\mathbf{s})}{\mathrm{len}(\mathbf{s})} < T$$

If a sentence has no repeat patterns at all, then the length of the codebook should be $C\mathrm{len}(\mathbf{s})$, and $\mathrm{MDL}(\mathbf{s}) \geq C\mathrm{len}(\mathbf{s})$. Thus we choose $T = C$.

For the NLLB mined corpus, we remove pairs with laser score < 1.06 or language score < 0.95 provided by LASER. Monolingual texts are cleaned using language scores only.

Table 1 and Figure 1 summarize the size of our training data after data cleaning and deduplication.

2.3 Preprocessing and Post Processing

There are thousands of languages in the world, thus statically training a tokenizer on a predefined list of languages is not flexible for new languages. There are several studies on dynamic vocabulary for new language adaption, the general principle is to maximize the overlap with the old vocabulary. (Lakew et al., 2018, 2019)

Source	Sentence Pairs
Constrained Track	$50.5M$
NLLB	$29.1M$
Self Collected	$151.6M$
Back Translation	$1.41B$
Total	$1.64B$

Table 1: Number of sentence pairs from different sources after data cleaning.

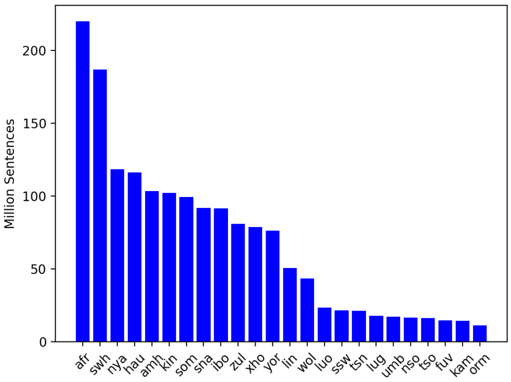


Figure 1: Number of sentences (in millions) in different African languages after data cleaning.

We reuse the mRASP2 tokenizer, a unigram model trained on 150 languages using SentencePiece (Pan et al., 2021). To support new African languages, we train another tokenizer for new languages and merge it to the mRASP2 tokenizer. To ensure that the merged tokenizer produces the same segmentation for old languages, new words that can be made by joining two or more old words are removed and the rest new words' probabilities are scaled down.

We notice that the Yoruba text in FLORES200 has more accented characters than other corpora. According to NLLB team's report, the way FLORES200 marks the tone of vowels is similar to MAFAND dataset (Adelani et al., 2022a). Thus, we use the MAFAND data to train an accent model to post-process the translated sentences for $X \rightarrow$ Yoruba translation. It takes the Yoruba character sequence with accents removed as the input and outputs the accented characters. The structure of the model is a two-layer bidirectional LSTM having 50 hidden units in each layer. Correspondingly, we train another accent model using non-MAFAND datasets to preprocess source text for Yoruba$\rightarrow X$ translation.

3 Model

3.1 Model Architecture

Existing research works demonstrate that small models suffer from the underfitting problem for multilingual machine translation. On the other hand, training and serving large models are expensive. Sometimes model parallelism or pipeline parallelism is necessary if it is impossible to run training on a single GPU due to memory constraints. And quantization is required to reduce the latency of inference. Our compromised model is a pre-layer norm transformer with $2.1B$ parameters which can be trained using $A100$ GPUs with $80G$ memory without parallelism. Details of the model are described in Table 2

Parameter	Value
Encoder Layer	64
Decoder Layer	64
Hidden Size	1024
FFN dimension	4096
Max Length	512
Shared Embedding	Decoder input output
Positional Embedding	Learned

Table 2: Architecture of our transformer model

3.2 Language Tag

There are two popular language tag strategies for multilingual MT: S-ENC-T-DEC which adds source language token to encoder input and target language token to decoder input (Fan et al., 2021; Liu et al., 2020; Wu et al., 2021), and T-ENC which adds target language token to encoder input (Yang et al., 2021; Wu et al., 2021). Our system uses T-ENC-T-DEC which adds the target language token to both encoder and decoder inputs. We did not use source language information for two reasons. First, most translation engines detect input languages automatically, which may introduce incorrect source language tokens. Second, a source sentence may be written in mixed languages.

4 Training and Optimization

4.1 Platform

Our models are trained on 6 machines each equipped with 8 Nvidia $A100$ $80G$ GPUs. We use our internal version of ParaGen [9] (Feng et al., 2022) , a self-developed text generation framework, to train the model. For back-translation, monolingual data are split and translated in parallel using 50 Nvidia Tesla $V100$ GPUs.

To accelerate training, LightSeq is integrated. Unlike approaches that proposed alternative model structures to trade quality for speed, LightSeq used a series of GPU optimization techniques tailored to the specific computation flow and memory access patterns of transformer models. It has been demonstrated 50% to 250% faster than Apex [10] on machine translation tasks. (Wang et al., 2021, 2022) Its inference speed is about 11.5 sentences per second using a single Nvidia Tesla $V100$ GPU, which allows us to translate all monolingual texts within a month.

As the training set's size exceeds the local disk's capacity, it is stored on a remote Hadoop file system.

4.2 Hyper-parameter Tuning

We tune the hyperparameters using a hill climbing approach where each iteration searches along one direction with a different value in the hyperparameter space while keeping the others constant in order to converge to the locally optimal solution on the validation set. To search efficiently, we fix a small batch size and tune other parameters, then increase the batch size after the other parameters have been tuned.

The final configuration is listed in Table 3.

Hyperparameter	Value
Initial Learning Rate	0.001
Warmup Steps	1000
Learning Rate Scheduler	Inverse Square Root
Dropout Rate	0.1
Sampling Temperature	5
Label Smoothing	0.1
Optimizer	AdamW$(0.9, 0.98)$
Activation Function	ReLU
Batch Size	$21M$ tokens

Table 3: Hyperparameters for training.

4.3 Streaming Data Shuffling

Data Shuffling reduces the variance of mini-batches and lowers the risk of local optimum. However, it is challenging to shuffle a Terabyte-scale dataset

[9] https://github.com/bytedance/ParaGen

[10] https://github.com/NVIDIA/apex

dynamically. Our system uses multi-source streaming data based shuffling, which maintains a small in-memory buffer and a set of file pointers that point to random offsets of the training set. Each time a file pointer is selected randomly and loads the next sample to the buffer. A batch of samples is drawn from the buffer randomly once the buffer is full. This approach takes the advantage of data prefetching for sequential access in the Hadoop file system. The randomness of the sampling is controlled by the number of file pointers and the size of the buffer. In our experiments, we use about $5k$ file pointers and $300G$ host memory for the buffer.

To compare with global dynamic shuffling, we run a simulation experiment. We train a model until convergence, then shuffle the full dataset statically, and continue training on the shuffled data. Repeat shuffling until no significant change in loss or performance. For clarity, the original model is named as M_0, and the model trained with $i-th$ round of shuffled data is named as M_i.

Table 4 shows the averaged per token loss of the last 100 training steps and averaged BLEU of M_i on English ↔ African language translations. We observed a slight improvement in the first round, but no significant change in the second round. This experiment suggests that our shuffling method combined with a limited number of static shuffling is a good approximation of global dynamic shuffling.

	M_0	M_1	M_2
Averaged Loss	1.95	1.91	1.91
Averaged BLEU	21.39	21.48	21.49

Table 4: Simulation Experiment of global dynamic data shuffling: M_0 is the model trained on original training data. M_i is the model trained on the $i-th$ round of statically shuffled data using M_{i-1} as the initial point. The averaged training loss over the last 100 steps and averaged BLEU of English ↔ African translations are reported.

4.4 Small Dynamic Vocabulary vs Large Static Vocabulary

Existing studies on vocabulary size do not reach a consensus. Large vocabularies often outperform small ones (Gowda and May, 2020), but not always (Liao et al., 2021)

Our vocabulary has $100k$ words, smaller than most of the other systems. Another difference is that our vocabulary is incrementally built for more than 150 languages, it may miss important words in new languages.

To understand the impact of vocabulary, we train another large unigram model with $200K$ words on the 26 languages in this shared task. Table 5 shows the performance with different vocabularies. It is obvious that the $100K$ vocabulary outperforms the $200K$ vocabulary, about 0.3 improvement in BLEU on average.

Vocabulary Size	Languages	BLEU
$100k$ words	173	21.97
$200k$ words	26	21.64

Table 5: Average BLEU of English ↔ African translations on the FLORES200 devtest set for the models with different vocabularies.

4.5 Pivot vs Direct

As reported in Microsoft's work, pivot-based translation is more robust, especially for directions between low-resource languages since corpora of $X \leftrightarrow Y$ are commonly sparser than $X \leftrightarrow$ English. (Yang et al., 2021) Therefore we use English as the pivot language for African-African translation. For French-African translation, the size of $X \leftrightarrow$ French data is comparable to $X \leftrightarrow$ English. Thus, we train a model for both English and French and choose the better one during inference time.

4.6 Model Averaging

As suggested by other works, model averaging is a simple trick that could significantly improve the performance without changing the model structure or slowing the inference speed. The only cost is the external disk spaces to save intermediate checkpoints, which is trivial compared with GPU and memory costs.

We save the checkpoints every 100 updates of gradients and average the last K checkpoints. By enumerating K from 1 to 20, we find that $K = 10$ is large enough to capture most of the gains.

5 Results

5.1 System Tuning

We tune our model on the FLORES200 devtest dataset, starting with a base model trained on the official data for the constrained track. Then we add more datasets and apply the optimization described above to boost performance. Table 6 reports the averaged BLEU over 56 directions including 24 African languages from and to English and 4 African languages from and to French.

Model Description	BLEU
Base model	16.92
+ NLLB and self-collected data	18.89
+ Data cleaning	19.64
+ Back-translation data	22.85
+ X → English → French	22.95
+ French → English → X †	22.90
+ Yoruba Accent for X → Yoruba	23.20
+ Yoruba Accent for Yoruba → X †	23.17
+ Model Averaging	23.35

Table 6: System tuning on FLORES200 devtest set, averaged BLEU over 56 directions is reported. Superscript † means the modification is not included in the final submission.

We can see that the amount of training data is proportional to the performance of the model, especially when back-translation data is added. For some very low resource languages such as Wolof, back-translation improves Wolof → English from 11.1 to 19.3, and English → Wolof from 4.17 to 7.07.

Another observation is that pivot translation outperforms direct translation for X → French directions, but underperforms for French → X, which indicates that the final step in pivot translation dominates the overall performance.

The impact of Yoruba accent models also shows mixed results. There is a significant improvement for X → Yoruba translation, but a little damage to Yoruba→ X translation. One possible reason is that the non-MAFAND dataset has multiple sources with different accent annotation standards, making the accent model confused. Therefore we only apply post-processing for X → Yoruba translations.

5.2 Final Result

Official evaluation metrics include BLEU, sentence-piece BLEU (spBLEU) score, and chrF++. Table 7 shows the results of our primary submission on FLORES200 dev, FLORES200 devtest set, and hidden test sets respectively. The sentence-piece model for calculating spBLEU is SPM-200 provided by Meta AI [11]

Dataset	BLEU	spBLEU	chrF++
FLORES dev	17.41	21.70	42.01
FLORES devtest	17.43	21.71	41.99
Official test	17.30	21.90	41.87

Table 7: Results of our primary submission on FLORES200 dev, FLORES200 devtest and official test datasets respectively. Metrics are averaged over 100 language pairs.

6 Conclusion

This paper presents our system for the WMT22 shared task on Multilingual Machine Translation for African Languages. We focus on data collection, augmentation, and cleaning. Due to the limited time, we did not try modeling tricks such as reranking and ensemble. Our finding is that the amount of data is crucial for translation quality, especially monolingual data in low-resource languages.

Acknowledgements

We thank Ying Xiong and Yang Wei for building the LightSeq package for this submission. We also thank the GMU-eval team for their effort to make our system work on the evaluation platform.

References

Common crawl. https://commoncrawl.org/. Accessed: 2022-07-18.

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. 2022. Towards a Cleaner Document-Oriented Multilingual Crawled Corpus. *arXiv e-prints*, page arXiv:2201.06642.

David Adelani, Jesujoba Alabi, Angela Fan, Julia Kreutzer, Xiaoyu Shen, Machel Reid, Dana Ruiter, Dietrich Klakow, Peter Nabende, Ernie Chang, Tajuddeen Gwadabe, Freshia Sackey, Bonaventure F. P. Dossou, Chris Emezue, Colin Leong, Michael Beukman, Shamsuddeen Muhammad, Guyo Jarso, Oreen Yousuf, Andre Niyongabo Rubungo, Gilles Hacheme, Eric Peter Wairagala, Muhammad Umair Nasir, Benjamin Ajibade, Tunde Ajayi, Yvonne Gitau, Jade Abbott, Mohamed Ahmed, Millicent Ochieng, Anuoluwapo Aremu, Perez Ogayo, Jonathan Mukiibi, Fatoumata Ouoba Kabore, Godson Kalipe, Derguene Mbaye, Allahsera Auguste Tapo, Victoire Memdjokam Koagne, Edwin Munkoh-Buabeng, Valencia Wagner, Idris Abdulmumin, Ayodele Awokoya, Happy Buzaaba, Blessing Sibanda, Andiswa Bukula, and Sam Manthalu. 2022a. A few thousand translations go a long way! leveraging pre-trained models for African news translation. In *Proceedings of*

[11] https://github.com/facebookresearch/fairseq/tree/nllb

the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3053–3070, Seattle, United States. Association for Computational Linguistics.

David Ifeoluwa Adelani, Md Mahfuz Ibn Alam, Antonios Anastasopoulos, Akshita Bhagia, Marta Costa-Jussá, Jesse Dodge, Fahim Faisal, Christian Federmann, Natalia Fedorova, Francisco Guzmán, Sergey Koshelev, Jean Maillard, Vukosi Marivate, Jonathan Mbuya, Safiyyah Saleem, and Holger Schwenk. 2022b. Findings of the WMT 2022 shared task on large-scale machine translation evaluation for african languages. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

Burton H. Bloom. 1970. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Jiangtao Feng, Yi Zhou, Jun Zhang, Xian Qian, Liwei Wu, Zhexi Zhang, Yanming Liu, Mingxuan Wang, Lei Li, and Hao Zhou. 2022. Paragen : A parallel generation toolkit.

Thamme Gowda and Jonathan May. 2020. Finding the optimal vocabulary size for neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3955–3964, Online. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Xiang Kong, Adithya Renduchintala, James Cross, Yuqing Tang, Jiatao Gu, and Xian Li. 2021. Multilingual neural machine translation with deep encoder and multiple shallow decoders. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1613–1624, Online. Association for Computational Linguistics.

Surafel M. Lakew, Aliia Erofeeva, Matteo Negri, Marcello Federico, and Marco Turchi. 2018. Transfer learning in multilingual neural machine translation with dynamic vocabulary. In *Proceedings of the 15th International Conference on Spoken Language Translation*, pages 54–61, Brussels. International Conference on Spoken Language Translation.

Surafel M. Lakew, Alina Karakanta, Marcello Federico, Matteo Negri, and Marco Turchi. 2019. Adapting multilingual neural machine translation to unseen languages. In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong. Association for Computational Linguistics.

Baohao Liao, Shahram Khadivi, and Sanjika Hewavitharana. 2021. Back-translation for large-scale multilingual machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 418–424, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8(0):726–742.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 244–258, Online. Association for Computational Linguistics.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*, pages 157–167. Association for Computational Linguistics.

Brian Thompson and Philipp Koehn. 2019. Vecalign: Improved sentence alignment in linear time and space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the*

9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 1342–1348, Hong Kong, China. Association for Computational Linguistics.

Xiaohui Wang, Yang Wei, Ying Xiong, Guyue Huang, Xian Qian, Yufei Ding, Mingxuan Wang, and Lei Li. 2022. Lightseq2: Accelerated training for transformer-based models on gpus. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '22, Dallas, Texas. Association for Computing Machinery.

Xiaohui Wang, Ying Xiong, Yang Wei, Mingxuan Wang, and Lei Li. 2021. LightSeq: A high performance inference library for transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 113–120, Online. Association for Computational Linguistics.

Liwei Wu, Shanbo Cheng, Mingxuan Wang, and Lei Li. 2021. Language tags matter for zero-shot neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3001–3007, Online. Association for Computational Linguistics.

Jian Yang, Shuming Ma, Haoyang Huang, Dongdong Zhang, Li Dong, Shaohan Huang, Alexandre Muzio, Saksham Singhal, Hany Hassan, Xia Song, and Furu Wei. 2021. Multilingual machine translation systems from Microsoft for WMT21 shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 446–455, Online. Association for Computational Linguistics.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

WebCrawl African : A Multilingual Parallel Corpora for African Languages

Pavanpankaj Vegi, Sivabhavani J, Biswajit Paul, Abhinav Mishra, Prashant Banjare, Prasanna Kumar KR, Chitra Viswanathan
pavanpankaj333@gmail.com, {jsbhavani, biswajit, abhinavmishra}.cair@gov.in,
{prashantban408, prasanna, chitrav}.cair@gov.in
Centre for Artificial Intelligence and Robotics, CV Raman Nagar, Bangalore

Abstract

WebCrawl African is a mixed domain multilingual parallel corpora for a pool of African languages compiled by ANVITA machine translation team of Centre for Artificial Intelligence and Robotics Lab, primarily for accelerating research on low-resource and extremely low-resource machine translation and is part of the submission to WMT 2022 shared task on Large-Scale Machine Translation Evaluation for African Languages under the data track. The corpora is compiled through web data mining and comprises 695K parallel sentences spanning 74 different language pairs from English and 15 African languages, many of which fall under low and extremely low resource categories. As a measure of corpora usefulness, a MNMT model for 24 African languages to English is trained by combining WebCrawl African corpora with existing corpus and evaluation on FLORES200 shows that inclusion of WebCrawl African corpora could improve BLEU score by 0.01-1.66 for 12 out of 15 African→English translation directions and even by 0.18-0.68 for the 4 out of 9 African→English translation directions which are not part of WebCrawl African corpora. WebCrawl African corpora includes more parallel sentences for many language pairs in comparison to OPUS public repository. This data description paper captures creation of corpora and results obtained along with datasheet. The WebCrawl African corpora is hosted on GitHub repository [1].

1 Introduction

Parallel corpus play a vital role in the progress of data driven machine translation research and development. Availability of parallel corpora is still a concern for a large collection of world languages. Africa alone is home to an estimated 1200 to 2100 spoken languages[2] and more than 34 languages with 1 Million plus speakers. Many of these 34 languages and associated language pairs fall under the low and extremely low resource categories and machine translation researchers face setbacks due to unavailability of parallel corpus in public domain.

WebCrawl African corpora is a little step put forward towards addressing this issue. Languages covered in WebCrawl African corpora include (1) Afrikaans(afr), (2) Amharic(amh), (3) Chichewa(nya), (4) Hausa(hau), (5) Igbo(ibo), (6) Lingala(lin), (7) Luganda(lug), (8) Oromo(orm), (9) Swahili(swh), (10) Swati(ssw), (11) Tswana/Setswana(tsn), (12) Xhosa(xho), (13) Xitsonga(tso), (14) Yoruba(yor) (15) Zulu(zul) and (16) English and language pairs include African-English and African-African pairs. WebCrawl African is submitted as a part of Large-Scale Machine Translation Evaluation for African Languages shared task(data track) of WMT22 Adelani et al. (2022).

Rest of the paper is organized as follows. Section-2 briefly covers related work on parallel corpora compilation through web data mining. Section-3 covers content collection process followed for WebCrawl African corpora compilation, Section-4 details its alignment processes, Section-5 presents results and analysis. Finally Section-6 presents the datasheet capturing responses to bunch of critical questions capturing many relevant facets of WebCrawl African corpora ranging from motivation, composition, collection process, processing, users, distribution, maintenance and Section-7 conclusion.

2 Related Work

A good amount of translated text are available on the web. However compilation of parallel corpora from web which involves suitable source discovery, sentence extraction, sentence alignment and quality assessment, control is not trivial. Sentence

[1] https://github.com/pavanpankaj/Web-Crawl-African
[2] https://en.wikipedia.org/wiki/Languages_of_Africa

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1076 - 1089
December 7-8, 2022 ©2022 Association for Computational Linguistics

alignment is the most critical part and alignment techniques range from simple heuristics to neural sentence embedding. Bañón et al. (2020) compiled ParaCrawl corpora from selected websites comprising of 41 languages and Vec/Hun/BLEU-Aligned techniques were used for sentence alignment. Schwenk et al. (2021a) compiled WikiMatrix corpora from Wikipedia articles comprising of 85 languages and used cross-lingual LASER embeddings, distance based measures and FAISS library for fast sentence alignment. Schwenk et al. (2021b) created CCMatrix corpora from snapshots of CommonCrawl comprising of 137 languages and used cross-lingual LASER embeddings, distance based measures, FAISS library and vector compression for fast, storage efficient sentence alignment. Ramesh et al. (2022) compiled Samanantar corpora from selected websites comprising of 11 Indian languages and used LaBSE cross-lingual embeddings, cosine similarity and FAISS library for fast sentence alignment. Philip et al. (2021) proposed an iterative alignment-training-alignment method for expanding corpora of Indian languages.

3 Content Collection through Web Crawling

Creation of parallel corpora through web data mining, by making use of sources of multilingual translated text present on the web has almost became the de-facto technique for its cost effectiveness and scaling advantages. WebCrawl African corpora creation followed similar strategy. As a first step, search has been carried out to discover potential websites having the following characteristics.

- Source website preferably should comprise of large number of text articles published in more than one African languages or/and English.
- Source website should have permissive Copyright T&C and favourable content usage policy.
- Source website should aid in covering diverse information domains, writing styles, genre and contains text covering contemporary language usage etc.
- Source website should have reasonable credibility for ensuring content quality in terms how contents are populated, content review mechanism followed, chances of biases of various forms in hosted content etc.

We ended-up finding four websites namely (1) South African Government[3] comprising of Government communication, (2) Nalibali [4] comprising of multi-genre short stories, (3) Gotquestions[5] comprising of spiritual Q&A and (4) African gospel[6] comprising of song lyrics.

Text content is mined from these four identified websites following four step process.

- Analyze website layout and collect relevant content through suitable web crawler
- Preserve alignment supervision signals such as web-page/document level hyperlinks across languages etc, wherever available
- Extract plain text by stripping of html tags
- If script is latin then apply nltk English sentence tokenizer else manually check sentence delimiter and apply delimiter to tokenize sentences
- Further align sentences following alignment algorithms-1, 2, 3

A relative comparison of 4 websites in terms of their contributions to the WebCrawl African corpora is shown in Figure-1

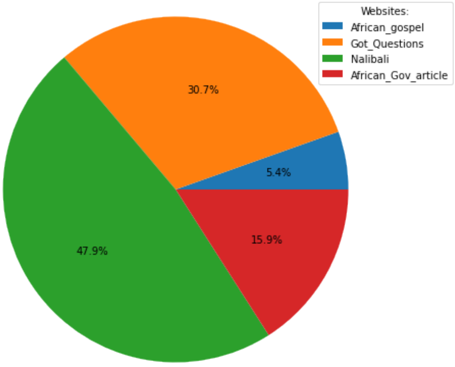


Figure 1: Source wise contributions in the WebCrawl African corpora

4 Alignment of Parallel Sentences

A good alignment strategy is expected to leverage alignment supervision signals available at the source websites. Since hyperlinks connecting

[3] https://www.gov.za/
[4] https://nalibali.org/
[5] https://www.gotquestions.org/
[6] https://africangospellyrics.com/

African and English language web-pages are available in the websites selected, the same is exploited for web-page level alignment and consequently search space for sentence alignment reduced significantly. Two different strategies are employed for sentence alignment duly leveraging the source websites information structure. Algorithm 1 is used for English-African parallel sentence alignment using cross-lingual embeddings and Algorithm 2 3 is used for fast African-African parallel sentence alignment based on common English sentences without using computationally expensive cross-lingual embeddings approach.

4.1 African-English Sentence Alignment

On an average, each web-page is having 200 to 250 sentences and hyperlinks to other language translated pages. Sentences from web-page aligned sources are extracted and segregated into source and target languages. Though web-pages are aligned, this unfortunately does not assure sentence level alignment due to improper sentence tokenization or even translation and format errors at the source. So a distinct need exists for carrying out sentence alignment exercise post segregation. Hence sentences are further aligned based on multilingual sentence encoders LASER[7] provided by the organizer of WMT22 Large-Scale Machine Translation Evaluation for African Languages shared task Adelani et al. (2022) and also heuristics. For a given row/sentence in source side, embeddings of all the target rows/sentences within a dynamic window around the source row is computed and the target row having maximum cosine similarity is selected as the source aligned sentence. Details are described in Algorithm 1. Time complexity of this African-English alignment algorithm depends on window-size. In worst case scenario, window-size can go up to number of source/target sentences and time complexity O(n^2), where n is max(#source sentences, #target sentences). Typically for the web-page aligned sources used, #source sentences or #target sentences range from 200 to 250.

4.2 African-African Sentence Alignment

The strategy employed for aligning African-African parallel sentences utilizes aligned African-English parallel sentences and does a fast alignment based on common English sentences without utilizing expensive cross-lingual embeddings.

5 Results and Analysis

We propose to evaluate the compiled WebCrawl African corpora in three ways. First, we present the distribution of extracted parallel sentences across language pairs. We then assess its usefulness by training a MNMT system for 24 African→English directions and finally compare it with resources available on public domain like OPUS.

5.1 Statistics of WebCrawl African Corpora

WebCrawl multilingual parallel corpora comprises a total of 695K mixed domain parallel sentences distributed non-uniformly over 74 language pairs. The parallel sentences are mined from web-pages/documents such as government notifications, short stories, descriptive answers to spiritual questions and lyrics. The range of sentences varies from around 85 sentences (Hausa-Swati) to 64500 sentences (Swahili-English). For the monolingual corpora, the range varies from around 1,300 sentences for Igbo to 64,500 sentences for Swahili. Primary reason which influenced the number of parallel sentences is non-uniform coverage of text across languages on the websites sourced for the corpora compilation. Number of parallel sentences per language pair is captured in Table-1.

As per Table 1, African languages are relatively rich in vocabulary as compared to English. However this trend is not observed in case of Amharic, Hausa languages. Also its interesting to observe that even though Xhosa is not having the highest number of sentences, but has the highest vocabulary(xho-eng) among all pairs.

5.2 Usefulness of WebCrawl African Corpora

As a measure of corpora usefulness, two MNMT models for 24 African languages to English are trained. First one with the existing corpus and second one by combining WebCrawl African corpora with the existing corpus and both are evaluated on FLORES200 Costa-jussà et al. (2022). Results as shown in Table 2 show that inclusion of WebCrawl African corpora could improve BLEU score by 0.01-1.66 for 12 out of 15 African→English translation directions and even by 0.18-0.68 for the 4 out of 9 African→English translation directions which are not part of WebCrawl African corpora, in spite being a tiny fraction as compared to the existing corpus. Potential reason could be WebCrawl African provides complimentary data to that of available

[7] https://github.com/facebookresearch/LASER/tree/main/tasks/wmt22

Algorithm 1 Algorithm for sentence alignment: African-English languages

```
Input: Tokenized sentences of srcLang(srcSentTok) and tgtLang(tgtSentTok), language encoder provided by organizers(src-lang-encoder),tgt-lang-encoder
Output: Aligned senteces for $srcSentTok$
$nsrc \leftarrow len(srcSentTok)$ , $ntgt \leftarrow len(tgtSentTok)$
$windowSize \leftarrow abs(nsrc - ntgt) + 2$    ▷ abs: absolute value, 2 is added to windowSize as an additional margin to error i.e tokenization error, translator error
$i = 0$
while $i > nsrc$ do
    if $i - windowSize > 0$ and $i + windowSize < nsrc$ then
        $windowSent \leftarrow tgtSentTok[i - windowSize : i + windowSize]$
    else if $i - windowSize > 0$ and $i + windowSize >= ntgt$ then
        $windowSent \leftarrow tgtSentTok[i - windowSize : ntgt]$
    else if $i - windowSize < 0$ and $i + windowSize <= ntgt$ then
        $windowSent \leftarrow tgtSentTok[0 : i + windowSize]$
    else if $i - windowSize < 0$ and $i + windowSize >= ntgt$ then
        $windowSent \leftarrow tgtSentTok[0 : ntgt]$
    end if
    compute vector embedding of $srcEmbed[i] \leftarrow srcLangEncoder(i)$
    for all $j \in windowSent$ do
        compute vector embedding of $windowEmbed[j] \leftarrow tgtLangEncoder(j)$
        compute similarity $cosSimScore[j] \leftarrow cosine\_sim(srcEmbed[i], windowEmbed[j])$
    end for
    $maxind \leftarrow indexofmax(cosSimScore)$
    Required aligned sentence is $srcSentTok[i]$ with $windowSent[maxind]$
    $i = i + 1$
end while
```

Algorithm 2 Algorithm for sentence extraction/alignment: African-African languages

```
Input: parallel sentences of African-lang, English and Other-African-lang, English sentence pairs of all articles
Output: African-African-lang-p-sent, Other-African-African-lang-p-sent
j=0
while $j < len(articles)$ do
    sentence pairs in $j^{th}$ article African-lang-en-p-sent,eng-p-sent and Other-African-lang-en-p-sent,eng-other-p-sent(afr-en pairs extracted from 1)
    Matching_indices = Compute_intersection(eng-p-sent, eng-other-p-sent)
    Align African-African-lang-p-sent, Other-African-African-lang-p-sent based on Matching_indices
    $African - African - lang - p - sent, Other - African - African - lang - p - sent$ are required gold parallel sentence pairs
    $j = j + 1$
end while
```

Algorithm 3 Algorithm for Compute_intersection

```
Input: Sentences of eng-p-sent, Sentences of eng-other-p-sent
edit_threshold = 4
Output: Returns list of tuples. for example [(1,1),(2,4)..] , means edit_distance(eng-p-sent[2], eng-other-p-sent[4]) <=edit_threshold
index1=0
index2=0
while $index1 < len(eng - p - sent)$ do
    while $index2 < len(eng - other - p - sent)$ do
        if edit_distance($eng-p-sent[index1], eng-other-p-sent[index2]) < edit\_threshold$ then
            tuple1 = $(index1, index2)$
            Matching_indices.append(tuple1)
        end if
        $index2 = index2 + 1$
    end while
    $index1 = index1 + 1$
end while
Return Matching_indices
```

in the existing corpus. Both the experiments used identical parameters and corpora used are only the only difference. For training, both WMT22 and WebCrawlAfrican+WMT22 corpus are further filtered using heuristics: (1) either source or target sentence is empty, (2) either source or target sentence length greater than 800 characters, (3) length of source and target sentence ratio is greater than 2.5 or length of source and target sentence ratio is less than 0.4 and (4) source or target sentence contains word having length greater than 10, (5) source or target sentence length is less than 4 and (6) source and target sentences are equal. Transformer with 24 layers of encoder and 6 layers of decoder are used for training both the models.

5.3 Comparison of WebCrawl African with OPUS Repository

A large part of African languages fall under the low and extremely low resource categories and do not have availability of parallel corpus of reasonable size in the public domain. A comparison of WebCrawl African corpora is carried out with the publicly available African parallel corpus listed on OPUS[8] repository in terms of parallel sentences.

Comparison results as captured in Figure-2 shows that out of 15 African-English language pairs compared, WebCrawl-African corpora has more number of parallel sentences for 7 African-English language pairs namely Chichewa-English, Lingala-English, Luganda-English, Oromo-English, Swati-English, Tswana-English, Tsonga-English languages as compared to OPUS public repository at the time of writing this paper. In fact WebCrawl-African corpora has 4 languages namely Chichewa, Luganda, Swati, and Tswana for which OPUS repository doesn't have even a single parallel corpora with any languages. Same goes for a few other African-African language pairs as well.

5.4 Corpora Quality

Though parallel corpora using web data mining approach can be created at scale, controlling quality of such corpora throws a major challenge. Noises ranging from source side errors such as incorrect translation, misspelling, incorrect grammar, biases of various forms and processing errors such as improper sentence tokenization, sentence alignment, additions, deletions etc often are of concern. In case of WebCrawl African corpora, the first choice made is to source content from credible websites, where website content is mostly generated in a controlled manner and contents are further reviewed. Also since the sources used have aligned web-pages so extracted sentence qualities are expected to be relatively better.

However, the authors could not analyze the corpora for translation correctness, biases and other quality metrics due to lack of knowledge on African

[8] https://opus.nlpl.eu/

Table 1: Statistics of WebCrawl African Parallel Corpora. (a,b,c) values in each box represents: $a = sentence\_count * 1000$, $b = unique\_source\_tokens * 1000$ and $c = unique\_target\_tokens * 1000$)

SrcLang(↓), TgtLang(→)	**afr**	**amh**	**nya**	**eng**	**hau**	**ibo**	**lin**	**lug**	**orm**	**tsn**	**swh**	**ssw**	**xho**	**tso**	**yor**	**zul**
Afrikaans (afr)	-	2.537 4.875 0.877	0.955 2.492 3.956	62.2 41.956 30.936	2.591 4.663 4.545	0.155 0.613 0.824	0	2.068 4.582 7.848	3.338 5.599 9.669	18.753 22.104 19.433	13.680 14.612 22.979	10.994 18.468 41.630	33.465 27.243 72.769	19.681 22.327 19.116	3.071 5.157 5.120	32.813 26.647 68.533
Amharic (amh)	2.537 0.877 4.875	-	0.634 0.212 3.105	4.6 1.294 6.737	2.562 0.891 4.781	0.117 0.012 0.742	0	1.65 0.760 6.744	2.816 1.065 9.264	0	3.130 1.240 9.361	0.091 0.012 0.996	0	0	2.792 1.076 5.401	0
Chichewa (nya)	0.955 3.956 2.492	0.634 3.105 0.212	-	1.4 5.180 3.177	0.92 3.912 2.584	0.16 0.922 0.837	0	0.987 4.478 4.484	0.947 3.908 4.027	0	0.964 3.959 3.474	0.136 0.820 1.149	0	0	0.92 3.926 1.498	0
English (eng)	62.2 30.936 41.956	4.6 6.737 1.294	1.4 3.177 5.18	-	5.6 7.48 6.606	1.1 1.45 2.219	1.1 0.956 1.53	3.6 5.478 10.71	7.0 8.164 14.252	25.9 20.191 22.946	64.5 27.103 59.569	14.4 16.428 47.934	46.2 24.481 85.768	24.4 19.158 21.254	6.3 7.585 7.647	50.9 25.022 84.648
Hausa (hau)	2.591 4.545 4.663	2.562 4.781 0.891	0.920 2.584 3.912	5.6 6.606 7.480	-	0.122 0.729 0.727	0	2.175 4.623 8.060	3.896 5.603 10.394	0	3.747 5.574 10.006	0.085 0.613 0.935	0	0	4.152 5.943 6.444	0
Igbo (ibo)	0.155 0.824 0.613	0.117 0.742 0.012	0.169 0.837 0.922	1.1 2.219 1.450	0.122 0.727 0.729	-	0	0.168 0.861 0.955	0.161 0.805 1.003	0	0.174 0.854 0.864	0.119 0.694 1.084	0	0	0.142 0.797 0.500	0
Lingala (lin)	0	0	0	1.1 1.53 0.956	0	0	-	0	0	0	0	0	0	0	0	0
Luganda (lug)	2.068 7.848 4.582	1.650 6.744 0.76	0.987 4.484 4.478	3.6 10.710 5.478	2.175 8.06 4.623	0.168 0.955 0.861	0	-	2.139 7.632 7.434	0	2.130 7.875 6.707	0.116 0.797 1.141	0	0	2.266 8.235 4.507	0
Oroma (orm)	3.338 9.669 5.599	2.816 9.264 1.065	0.947 4.027 3.908	7.0 14.252 8.164	3.896 10.394 5.603	0.161 1.003 0.805	0	2.139 7.434 7.632	-	0	4.583 11.654 11.437	0.123 0.820 1.069	0	0	4.333 10.966 6.477	0 25.022 84.648
Tswana/Setswana (tsn)	18.753 19.433 22.104	0	0	25.9 22.946 20.191	0	0	0	0	0	-	0	11.14 15.779 41.229	19.694 19.455 55.865	19.442 19.533 19.052	0	18.904 19.393 52.589
Swahili (swh)	13.68 22.979 14.612	3.13 9.361 1.24	0.964 3.474 3.959	64.5 59.569 27.103	3.747 10.006 5.574	0.174 0.864 0.854	0	2.13 6.707 7.875	4.583 11.437 11.654	0	-	0.133 0.737 1.194	0	0	4.134 10.725 6.309	0
Swati (ssw)	10.994 41.63 18.468	0.091 0.996 0.012	0.136 1.149 0.82	14.4 47.934 16.428	0.085 0.935 0.613	0.119 1.084 0.694	0	0.116 1.141 0.797	0.123 1.069 0.82	11.140 41.229 15.779	0.133 1.194 0.737	-	11.274 40.769 41.968	11.515 42.138 15.236	0.118 1.144 0.462	11.139 41.609 40.488
Xhosa (xho)	33.465 72.769 27.243	0	0	46.2 85.768 24.481	0	0	0	0	0	19.694 55.865 19.455	0	11.274 41.968 40.769	-	20.449 56.629 19.272	0	33.638 72.821 68.472
Xitsonga (tso)	19.681 19.116 22.327	0	0	24.4 21.254 19.158	0	0	0	0	0	19.442 19.052 19.533	0	11.515 15.236 42.138	20.449 19.272 56.629	-	0	20.342 19.390 53.702
Yoruba (yor)	3.071 5.12 5.157	2.792 5.401 1.076	0.920 1.498 3.926	6.3 7.647 7.585	4.152 6.444 5.943	0.142	0	2.266 4.507 8.235	4.333 6.477 10.966	0	4.134 6.309 10.725	0.118 0.462 1.144	0	0	-	0
Zulu (zul)	32.813 68.533 26.647	0	0	50.9 84.648 25.022	0	0	0	0	0	18.904 52.589 19.393	0	11.139 40.488 41.609	33.638 68.472 72.821	20.342 53.702 19.39	0	-
Total(sentences)	206.3	20.92	8.032	319.7	25.85	2.427	1.1	17.299	29.336	113.833	97.175	71.383	164.72	115.829	28.228	167.736

languages. Further human evaluation by language experts could not be carried out due to shortage of time and resources. As far as diversity of domains, genre, writing style and contemporary use of languages are concern, the source websites selected are expected to address them reasonably well. The details are covered in the datasheet presented in the next section.

6 Datasheets for WebCrawl African Corpora

Toward the growing consensus of having systematic dissipation of information on dataset to all its stakeholders by capturing all relevant facets, we followed Gebru et al. (2021) the idea of datasheets for datasets and further its adaptation by Costa-jussà et al. (2020) for MT taks. Datasheet for the WebCrawl African corpora is given below.

6.1 Motivation

(a) Who created the dataset(e.g., which team, research group) and on behalf of which entity (e.g. company, institution, organization)?

WebCrawl African corpora is compiled by ANVITA machine translation team of Centre for Artificial Intelligence and Robotics Lab based in Bangalore.

(b)Did they fund it themselves? If there is an associated grant, please provide the name of the grantor and the grant name and number

WebCrawl African corpora compilation work is fully supported by the Centre for Artificial Intelligence and Robotics Lab. No external grants are received or used for this work.

(c) For what purpose was the data set created?

African→English	WMT22* (#sentence)	WMT22*+ WebCrawlAfrican* (#sentence)	WMT22* (95M) (BLEU)	WMT22* (95M) (CHRF2++)	WMT22*+ WebCrawlAfrican*(260K) (BLEU)	WMT22*+ WebCrawlAfrican*(260K) (CHRF2++)
afr-en	12128497	12179628	**55.8**	74.185	55.73	**74.21**
amh-en	946778	950103	**24.39**	48.80	24.17	**48.82**
nya-en	1415637	1417004	22.45	**48.79**	**22.66**	45.46
hau-en	3349586	3354753	27.92	49.95	**28.04**	**50.18**
ibo-en	372787	373452	20.62	44.07	**21.25**	**44.44**
kam-en	1452332	1452332	9.24	28.26	**9.49**	**28.33**
kin-en	8595328	8595328	25.97	48.01	**26.15**	**48.34**
lin-en	2294855	2295671	19.34	40.80	**19.56**	**41.2**
lug-en	2667772	2670662	15.93	37.09	**16.69**	**37.73**
luo-en	2339916	2339916	**17.34**	**38.51**	16.96	38.32
fuv-en	1256816	1256816	5.62	21.91	**5.82**	**21.95**
nso-en	2284885	2284885	33.30	53.77	**33.54**	**54.52**
orm-en	2139879	2145917	11.27	31.55	**12.13**	**33.57**
sna-en	7335877	7335877	**23.68**	46.43	23.57	**46.73**
som-en	1084345	1084345	**18.01**	**40.02**	17.80	40.02
swh-en	28152884	28208419	41.01	62.23	**41.19**	**62.32**
ssw-en	93532	105225	23.68	45.79	**25.34**	**47.27**
tsn-en	4257859	4278691	22.66	44.97	**23.08**	**45.96**
umb-en	247063	247063	**5.74**	**24.35**	5.55	24.27
wol-en	138994	138994	**8.71**	**27.10**	8.43	27.01
xho-en	7552496	7588334	31.8	53.78	**32.01**	**53.84**
tso-en	511184	531823	**24.32**	**45.85**	21.72	44.33
yor-en	1471404	1477092	15.38	37.12	**15.39**	**37.20**
zul-en	3352155	3355480	33.4	55.52	**33.79**	**55.70**

Table 2: MT performance (BLEU, CHRF2++) with and without WebCrawl African Corpora.[*] Filtered

Was there a specific task in mind? If so, please specify the result type (e.g. unit) to be expected

WebCrawl African corpora is created primarily for accelerating research on low resource and extremely low resource machine translation. This corpora is also part of the submission to WMT 2022 shared task on Large-Scale Machine Translation Evaluation for African Languages under data track.

(d) Could any of these uses, or their results, interfere with human will or communicate a false reality?

No such thing is communicated to the authors. However, as machine translation is not free from biases, errors and may fail to portray actual essence of the translation or portray false, unfair realities, so such things can not be ruled out for WebCrawl African corpora and its usage as well.

(e) What is the antiquity of the file? Provide, please, the current date. The first version of WebCrawl African corpora was released on 10 May 2022. There was no further release till the time of writing this response.

(f) Has there been any monetary profit from the creation of this dataset?

The dataset is created and released mainly to aid research in MT and hoping to be useful for other NLP research as well. It's not for any monetary profit in the past, present and future as well.

6.2 Composition

(a) Is there any synthetic data in the dataset? If so, in what percentage?

WebCrawl African corpora does not contain any synthetic data.

(b) Are there multiple types of instances or is there just one type? Please specify the type(s), e.g. Raw data, preprocessed, symbolic.

WebCrawl African corpora comprises 695K parallel sentences spanning 74 language pairs from 15 African languages and English. African languages covered include Afrikaans(afr), Lingala(lin), Swati(ssw), Amharic(amh), Luganda(lug), Tswana/Setswana(tsn), Chichewa(nya), Hausa(hau), Oroma(orm), Xhosa(xho), Igbo(ibo), Xitsonga(tso), Yoruba(yor), Swahili(swh), and Zulu(zul). Source and Target parallel sentences are part of two separate files having the following naming convention.

Source file : webcrawl-african-{src-lang}-{tgt-lang}.{src-lang}

Target file : webcrawl-african-{src-lang}-{tgt-

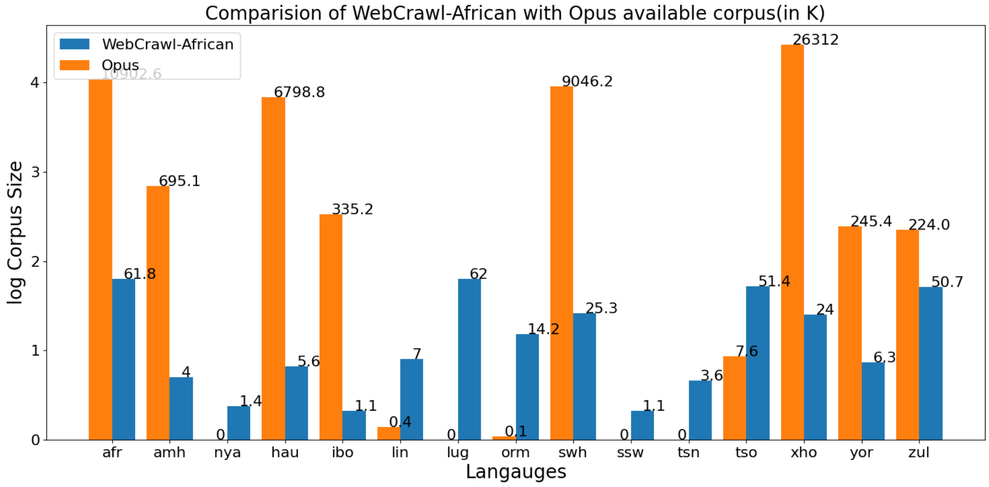


Figure 2: Comparison of WebCrawl-African corpora with the parallel corpus listed on OPUS repository

lang}.{tgt-lang}

src-lang and *tgt-lang* languages correspond to one of the 15 African languages and English part of WebCrawl African corpora and the whole corpora is spread over 148 files in 2 directories.

Monolingual corpora for language *src-lang* is available at *webcrawl-african-{src-lang}-eng.{src-lang}* file.

(c) What do the instances (of each type, if appropriate) that comprise the data set represent? (e.g. documents, photos, people, countries).

Instances represent parallel sentences aligned between two languages and stored in source and target files, following the naming convention mentioned above.

(d) How many instances (of each type, if appropriate) are there in total?

WebCrawl African parallel corpora comprises a total of 695K sentences (instances) distributed non-uniformly over 74 language pairs from 15 African languages and English. The range of sentences varies from around 85 sentences (Hausa-Swati) to 64,500 sentences (Swahili-English).

For the monolingual corpora available, the range of sentences varies from around 1,300 for Igbo to 64,500 sentences for Swahili. Complete count for each language pairs is available at the corpora hosting page `https://github.com/pavanpankaj/Web-Crawl-African`.

(e) Does the dataset contain all possible instances or is it just a sample of a larger set? i.e. Is the dataset different than an original one due to the preprocessing process? In case this dataset is a subset of another one, is the original dataset available?

WebCrawl African corpora is compiled by mining text from websites mentioned, through crawling and following sentence alignment techniques. Therefore, although the corpora is not a subset of any other corpora, it is limited by the text content crawled till the date of released of this corpora.

(f) Is there a label or a target associated with each of the instances? If so, please provide a description.

For any given language pair, a sentence in line number *i* and *Source language file : webcrawl-african-{src-lang}-{tgt-lang}.{src-lang}* will have an aligned target sentence in line number *i* and *Target language file : webcrawl-african-{src-lang}-{tgt-lang}.{tgt-lang}*. There are no other explicit labels associate with instances.

(g) What is the format of the data? e.g. .json, .xml, .csv .

For all language pairs, the aligned source and target sentences are kept in two seperate files following naming conventions mentioned in Section 6.2.(b) and all files are in UTF-8 plain text format.

(h) Is any information missing from individual instances? If so, please provide a description, explaining why this information is missing (e.g. because it was unavailable). This does not include intentionally removed information, but might include, e.g. redacted text.

No such thing is reported. However, due to the automated techniques employed for corpora creation, some sentences may have missing words. Also there are language pairs for which no parallel sentences are present, for example Lingala does not have any language pairs with all other African languages included in WebCrawl African corpora.

(i) Are there any errors, sources of noise, or redundancies in the dataset? If so, please provide a description. Do not include missing information here.

No such thing is reported. However, due to the automated techniques employed for corpora creation, sentence misalignment error and misalignment induced noises in small proportion can not be ruled out. Additionally, content collected from African Gospel lyrics website where the content is generated through crowdsourcing with not so strict content review mechanism may have noises ranging from misspelling, grammatical errors and use of informal writings.

(j) Is there any verification that guarantees there is not institutionalization of unfair biases? Both regarding the dataset itself and the potential algorithms that could use it.

No such study is carried out or mechanism employed to assess and address corpora biases. Corpora is compiled by mining text from websites mentioned and inherited biases can not be ruled out. So both WebCrawl African corpora and translation algorithms could present biases.

(k) Are there recommended data splits, e.g. training, development/validation, testing? If so, please provide a description of these splits explaining the rationale behind them.

No specific splits are recommended.

(l) Is the dataset self-contained, or does it link to or otherwise rely on external resources? e.g., websites, tweets, other datasets. If it links to or relies on external resources, i) Are there any guarantees that they will exist, and remain constant over time? ii) Are there official archival versions of the complete dataset? i.e. including the external resources as they existed at the time the dataset was created. iii) Are there any restrictions (e.g. licenses, fees) associated with any of the external resources that might apply to a future user? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, if appropriate.

WebCrawl African corpora is self-contained and hosting page as mentioned contains the complete corpora.

(m) Does the dataset contain data that might be considered confidential? e.g. data that is protected by legal privilege or by doctor patient confidentiality, data that includes the content of individuals non-public communications. If so, please provide a description.

Corpora is compiled by mining text available in the public domain. So such a presence is unlikely.

(n) Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? If so, please describe why.

Website content sourced for compiling corpora is meant for public consumption and genre includes government communication, short children stories, religious text and lyrics. So such anti-social content is unlikely. However, no review of the corpora is carried out from the perspective in question.

(o) Does the dataset relate to people? If so, please specify a) Whether the dataset identifies subpopulations or not. b) Whether the dataset identifies individual people or not. c) Whether it contains information that could vulnerate any individuals or their rights. c) Any other verified information on the topic that can be provided.

WebCrawl African corpora is compiled from open source content meant for public consumption and likely to reference people for the cause that made them appear publicly. Corpora does not include and express anything new which is not there in the public domain. However, no formal review of the corpora is carried out from the perspective in question.

(p) Does the dataset cover included languages equally?

Size of both parallel and monolingual corpora is not same for all the languages and language pairs included. Primary reason is the non-uniform coverage of text across languages on the websites sourced for the corpora compilation.

(q) Is there any evidence that the data may be somehow biased? i.e. towards gender, ethics, beliefs.

No study is carried out or mechanism employed to assess and address corpora biases. The corpora is compiled by mining text content available on the websites mentioned and inherited biases can not be ruled out.

(r) Is the data made up of formal text, informal text or both equitably?

WebCrawl African corpora comprises mostly formal text. However there are instances of informal content primarily coming from lyrics mined from African Gospel Lyrics website.

(s) Does the data contain incorrect language expressions on purpose? Does it contain slang terms? If that's the case, please provide which instances of the data correspond to these.

Given the genre of content hosted by the websites sourced for this corpora mining, such contents are unlikely. However, no review of the corpora is carried out from the perspective in question.

6.3 Collection Process

(a) Where was the data collected at? Please include as much detail; i.e. country, city, community, entity and so on.

WebCrawl African corpora is compiled by mining content hosted by websites (1) South African Government `https://www.gov.za/`, (2) Nalibali `https://nalibali.org/`, (3) Gotquestions `https://www.gotquestions.org/` and (4) African gospel `https://africangospellyrics.com/`. Websites comprise of text content covering government communication, multi-genre short stories, answers to spiritually related questions and gospel lyrics. A large part of it presumably written by the government officials, subject experts and volunteers primarily from the African countries and to some extent may be by the African speaking people from other countries. So data might be considered to have originated primarily from the African countries and other places around the globe as well. However, corpora compilation is carried out by the ANVITA team at Centre for Artificial Intelligence and Robotics, Bangalore.

(b) If the dataset is a sample from a larger set, what was the sampling strategy? i.e. deterministic, probabilistic with specific sampling probabilities.

WebCrawl African corpora is compiled by mining text content available on websites mentioned. The corpora is not a subset of any other corpora and no specific sampling was performed. However content is limited by the text crawled until the date of release of corpora.

(c) Are there any guarantees that the acquisition of the data did not violate any law or anyone's rights?

Websites having permissible copyright T&C and favourable content usage policy are used at the first place for content acquisition. Source websites permit usage and distribution of content for non-commercial, not-for-profit and fair use with due source acknowledgement. WebCrawl African corpora is hence released under CC-BY-NC-SA license for research purpose after intimation and with source acknowledgement. As long as WebCrawl African corpora license and source website copyright T&C and content usage policy is followed, one should safely assume that corpora acquisition and usage are unlikely to violate any laws or rights. Neither ANVITA team nor Centre for Artificial Intelligence and Robotics Lab holds any copyright over the WebCrawl African corpora. However, any derivatives of the corpora must acknowledge all sources including team ANVITA.

(d) Are there any guarantees that prove the data is reliable?

WebCrawl African corpora is created in an automated fashion without human verification like most of the large scale parallel corpora, thereby making it hard to guarantee provable reliability. However, as corpora is compiled by mining websites where content is mostly generated in a controlled manner and reviewed, makes the corpora reasonably reliable.

(e) Did the collection process involve the participation of individual people? If so, please report any information available regarding the following questions: Was the data collected from people directly? Did all the involved parts give their explicit consent? Is there any mechanism available to revoke this consent in the future, if desired?

As stated, content for the corpora compilation is directly sourced from websites mentioned and without direct participation of individual people.

(f) Has an analysis of the potential impact of the dataset and its use on data subjects been conducted? i.e. a data protection impact analysis. If so, please provide a description of this analysis, including the outcomes, as well as a link or other access point to any supporting documentation. Neither such analysis is conducted nor any communication received on the subject.

(g) Were any ethical review processes conducted?

No ethical review processes were conducted.

(h) Does the data come from a single source or is it the result of a combination of data coming from different sources? In any case, please provide references.

WebCrawl African corpora is compiled by mining content hosted by four websites (i) South African Government `https://www.gov.za/`, (ii) Nalibali `https://nalibali.org/`, (iii) Gotquestions `https://www.gotquestions.org/` and (iv) African gospel `https://africangospellyrics.com/`. Websites comprise of text content covering government communication, multi-genre short stories, answers to spiritually related questions and gospel lyrics and contributed in creating mixed domain corpora. However, compilation is carried out by the ANVITA team.

(i) If the same content was to be collected from a different source, would it be similar?

It's likely to be similar if the content collected has a similar topic, genre, writing style and content distributions.

(j) Please specify any other information regarding the collection process. i.e. Who collected the data, whether they were compensated or not, what mechanisms were used. Please, only include if verified.

Content acquisition and corpora compilation is carried out by ANVITA team using a four steps process.

(i) Identification of websites based on content coverage, copyright T&C, usage policy and credibility.

(ii) Analysis of website layout and collection of relevant content through crawling preserving webpage/document level alignment signals, wherever available.

(iii) Extraction of plain text by stripping of html tags and splitting of text at sentence level

(iv) Alignment of parallel sentences across language pairs

6.4 Processing/Cleaning/Labelling

(a) Please specify any information regarding the preprocessing that you may know (e.g. the person who created the dataset has somehow explained it) or be able to find (e.g. there exists and informational site). Please, only include if verified. i.e. Was there any mechanism applied to obtain a neutral language? Were all instances preprocessed the same way?

WebCrawl African corpora is available as sentence aligned files. Preprocessing steps on raw crawled web-pages include striping off html tags, sentence tokenization and sentence alignment. Sentence alignment was carried out based on cross-lingual embeddings using LASER encoder and heuristics to a large extent. The entire corpora is preprocessed in the same way. No word/subword/character level tokenization or other pre-processing like filtering on parallel sentences were carried out. However, further filtering based on heuristics similar to the one used by the authors for training MT models may be carried out for better performance.

6.5 Users

(a) Has the dataset been used already? If so, please provide a description.

WebCrawl African corpora is used as a resource for the WMT 2022 shared task on Large-Scale Machine Translation Evaluation for African Languages Adelani et al. (2022). This corpora is also used by the ANVITA team for training MT model and results are presented in this paper.

(b) When was the dataset first released?

The initial release of WebCrawl African corpora was 10 May 2022.

(c) Is there a repository that links to any or all papers or systems that use this dataset? If so, please provide a link or any other access point.

WMT event is supposed to present a findings paper for the 2022 edition that may include such reference. Also corpora hosting page is likely to maintain such repository.

(d) What (other) tasks could the dataset be used for? Please include your own intentions, if any.

WebCrawl African corpora is primarily intended for machine translation tasks. It can also be used as monolingual corpora for tasks such as language modeling, corpus based language studies and few other NLP tasks with additional annotations.

(e) Are there tasks for which the dataset should not be used? If so, please provide a description.

There is no explicit task where this corpora should not be used. However, use of WebCrawl African corpora is not recommended as a benchmark corpora.

(f) Any other comments? i.e. Do the collection or preprocessing processes impact future uses?)

Like any large parallel corpora, WebCrawl African corpora is created in an automated fashion without human verification.

6.6 Distribution

(a) Please specify the source where you got the dataset from.

As mentioned, WebCrawl African corpora is compiled by mining text from web-pages hosted by (i) South African Government https://www.gov.za/, (ii) Nalibali https://nalibali.org/, (iii) Gotquestions https://www.gotquestions.org/ and (iv) African gospel https://africangospellyrics.com/

(b) When was the dataset first released?

WebCrawl African corpora was first released on 10 May 2022.

(c) Are there any restrictions regarding the distribution and/or usage of this data in any particular geographic regions?

No, there are no such restrictions.

(d) Is the dataset distributed under a copyright or other intellectual property (IP) license? And/or under applicable terms of use (ToU)? Please cite a verified source.

WebCrawl African Corpora distributed under CC-BY-NC-SA license. Barring commercial use, the license allows mostly unrestricted fair usage.

(e) Any other comments? i.e. How has the data been distributed? Who has access to the dataset? When was the dataset first distributed? Are there any other regulations on the dataset?

WebCrawl African Corpora is distributed through GitHub public hosting at https://github.com/pavanpankaj/Web-Crawl-African and also WMT 2022 website at https://www.statmt.org/wmt22/large-scale-multilingual-translation-task.html since 10 May 2022 under CC-BY-NC-SA license.

6.7 Maintenance

(a) Is there any verified manner of contacting the creator of the dataset?

All queries on WebCrawl African corpora should be sent to Pavan Pankaj Vegi at pavanpankaj333@gmail.com and Biswajit Paul at biswajit.cair@gov.in.

(b) Specify any limitations there might be to contributing to the dataset. i.e. Can anyone contribute to it? Can someone do it at all?

Scope exists for extending the corpora with additional parallel sentences and language pairs, specifically involving low and extremely low resource languages. Contribution can be done by contacting ANVITA team members at pavanpankaj333@gmail.com and biswajit.cair@gov.in

(c) Has any erratum been notified?

No erratum has been notified.

(d) Is there any verified information on whether the dataset will be updated in any form in the future? Is someone in charge of checking if any of the data has become irrelevant throughout time? If so, will it be removed or labeled somehow?

WebCrawl African corpora is likely to be updated with additional parallel sentences in future by the ANVITA team. Though chances of corpora becoming irrelevant in near future are less likely, but if it happens, hosting page https://github.com/pavanpankaj/Web-Crawl-African will reflect the right status.

(e) Is there any available log about the changes performed previously in the dataset?

Not applicable, as the current version is the first version. However, log of future changes will be recorded at the corpora hosting page https://github.com/pavanpankaj/Web-Crawl-African

(f) Could changes to current legislation end the right-of-use of the dataset? WebCrawl African corpora is published under CC-BY-NC-SA license. We do not foresee any right-of-use changes in future.

(g) Any other comments? i.e. Is there someone supporting/hosting/maintaining the dataset? If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances?

Webcrawl African corpora is hosted at https://github.com/pavanpankaj/Web-Crawl-African and likely to be maintained by the ANVITA team..

7 Conclusion

This paper presented detailed description of WebCrawl African corpora. The paper also describes approach and design choices to systematically create parallel corpora and extend the WebCrawl African corpora through web data mining and alignment. WebCrawl African corpora compiled comprises 695K parallel sentences spanning 74 different language pairs from English and 15 African languages, many of which fall under low and extremely low resource categories. Webcrawl African corpora is hosted at https://github.com/pavanpankaj/Web-Crawl-African for non-commercial, not-for-profit and fair use. This corpora comprises sentences from multiple domains and includes government communication, short children stories, religious text and lyrics. Though human verification of the corpora was not carried out but favourable characteristics of selected source websites aided to address some of the quality concerns relatively better.

Experiments and evaluation of results show that inclusion of WebCrawl African corpora with WMT 2022 corpus has improved BLEU score by 0.01-1.66 for 12 out of 15 African→English translation directions and even by 0.18-0.68 for the 4 out of 9 African→English translation directions which are not part of WebCrawl African corpora and also it has more parallel sentences for many language pairs in comparison to OPUS public repository.

WebCrawl African corpora is primarily intended for machine translation tasks, specially for accelerating research on low resource and extremely low resource machine translation. It can also be used as monolingual corpora for tasks such as language modeling, corpus based language studies and few other NLP tasks with additional annotations.

Acknowledgements

The authors would like to thank Director, CAIR for his constant encouragement, guidance and enablement.

References

David Ifeoluwa Adelani, Md Mahfuz Ibn Alam, Antonios Anastasopoulos, Akshita Bhagia, Marta Costa-Jussá, Jesse Dodge, Fahim Faisal, Christian Federmann, Natalia Fedorova, Francisco Guzmán, Sergey Koshelev, Jean Maillard, Vukosi Marivate, Jonathan Mbuya, Safiyyah Saleem, and Holger Schwenk. 2022. Findings of the WMT 2022 hared task on large-scale machine translation evaluation for african languages. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Marta R Costa-jussà, Roger Creus, Oriol Domingo, Albert Domínguez, Miquel Escobar, Cayetana López, Marina Garcia, and Margarita Geleta. 2020. Mt-adapted datasheets for datasets: Template and repository. *arXiv preprint arXiv:2005.13156*.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92.

Jerin Philip, Shashank Siripragada, Vinay P Namboodiri, and CV Jawahar. 2021. Revisiting low resource status of indian languages in machine translation. In *8th ACM IKDD CODS and 26th COMAD*, pages 178–187.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Divyanshu Kakwani, Navneet Kumar, et al. 2022. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021a. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021b. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500, Online. Association for Computational Linguistics.

ANVITA-African: A Multilingual Neural Machine Translation System for African Languages

Pavanpankaj Vegi, Sivabhavani J, Biswajit Paul,
Prasanna Kumar KR, Chitra Viswanathan
pavanpankaj333@gmail.com, {jsbhavani.cair, biswajit.cair, prasanna.cair, chitrav.cair}@gov.in
Centre for Artificial Intelligence and Robotics, CV Raman Nagar, Bangalore, India

Abstract

This paper describes ANVITA African NMT system submitted by team ANVITA for WMT 2022 shared task on Large-Scale Machine Translation Evaluation for African Languages under the constrained translation track. The team participated in 24 African languages to English MT directions. For better handling of relatively low resource language pairs and effective transfer learning, models are trained in multilingual setting. Heuristic based corpus filtering is applied and it improved performance by 0.04-2.06 BLEU across 22 out of 24 African→English directions and also improved training time by 5x. Use of deep transformer with 24 layers of encoder and 6 layers of decoder significantly improved performance by 1.1-7.7 BLEU across all the 24 African→English directions compared to base transformer. For effective selection of source vocabulary in multilingual setting, joint and language wise vocabulary selection strategies are explored at the source side. Use of language wise vocabulary selection however did not consistently improve performance of low resource languages in comparison to joint vocabulary selection. Empirical results indicate that training using deep transformer with filtered corpora seems to be a better choice than using base transformer on the whole corpora both in terms of accuracy and training time.

1 Introduction

Africa is very rich in languages, and around 1200 to 2100 languages are spoken in African countries[1], 24 African languages and 100 language pairs were selected for the WMT22 Large-Scale Machine Translation Evaluation for African Languages shared task Adelani et al. (2022b). Selected 24 African languages include Afrikaans(afr), Amharic(amh), Chichewa(nya), Hausa(hau), Igbo(ibo), Kamba(kam), Kinyarawanda(kin), Lingala(lin), Luganda(lug), Luo(luo), Nigerian Fulfulde(fuv), Northern Sotho(nso), Oromo(orm), shona(sna), Somali(som), Swahili(swh), Swati(ssw), Setswana(tsn), Umbundu(umb), Wolof(wol), Xhosa(xho), Xitsonga(tso), Yoruba(yor) and Zulu(zul) and language pairs include African-English, selective African-French, and African-African pairs, where many of the pairs fall under the low resource category. In this task, organizers permitted two submissions, Best scoring submission is considered as Primary model and other one being the Contrastive model. This paper describes our submission to WMT 2022 Large-Scale Machine Translation Evaluation for African Languages shared task where we participated for translation of 24 African languages to English. We are not officially given a rank as we didn't participate in all African MT directions.

2 Related Work

Developing quality machine translation system for low resource languages still remains a major challenge and many of the world languages fall under this category. Some of the recent developments do show that multilingual NMT is a promising direction. In massively multilingual neural machine translation, the authors have shown to train a single model for translating 102 languages to and from English and the results outperformed the strong bilingual baseline MT system especially for low resource languages Johnson et al. (2017). However, it cannot be generalized to all high and medium resource languages. Gowda et al. (2021) built a multilingual neural machine translation system capable of translating from 500 source languages to English which includes medium, low and extremely low resource languages. Zhang et al. (2020a) improved zero-shot translation in multilingual neural machine translation by random back translation. Kudugunta et al. (2019) have shown that represen-

[1] https://en.wikipedia.org/wiki/Languages_of_Africa

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1090 - 1097
December 7-8, 2022 ©2022 Association for Computational Linguistics

tations of high resource and/or linguistically similar languages are more robust when fine-tuning on an arbitrary language pair, which is critical to determining how much cross-lingual transfer can be expected in a zero or few-shot setting.

Zhou et al. (2021) shown deep architectures for neural machine translation and post ensemble have shown improved results on machine translation tasks. Zhang et al. (2020b) presented language independent heuristics for filtering noisy pairs from parallel corpus. Yang et al. (2021) proposed progressive training, in which the MT system is trained from shallow to deep architectures - increasing number of encoder and decoder layers. However, major improvement is observed while increasing encoder layers. Adelani et al. (2022a) created novel African corpus for 16 African languages and finetuned on pre-trained large MT models. Fan et al. (2021) demonstrated massively multilingual machine translation by training a single model that can translate between any pair of 100 languages.

3 Datasets

We used all the parallel corpora provided by WMT 2022 organizer. Corpus contain existing OPUS repository Tiedemann (2012), WMT 2022 novel corpus[2] and comprises of sources such as wikimedia, CCMatrix, CCAligned, bible-uedin, GNOME, XLEnt, QED,KDE4, mozilla-I10n, SPC, TED2020, Tatoeba, ELRC_2922, OpenSubtitles, Ubuntu, LAVA corpus2, MAFAND-MT Adelani et al. (2022a), KenTrans Wanzare et al. (2022), Kencorpus McOnyango et al. (2022), WebCrawl-African[3] and huggingface (provided by organisers) etc. Tiedemann (2012). Combining all a total 140 Million parallel sentences for the 24 African-English language pairs are extracted.

Language wise statistics of corpus used in our system is listed in Table 1.

4 System Overview

ANVITA African MT system comprises of two major sub systems: Data preprocessing and Model training under different strategies and architectural configurations followed by evaluation.

4.1 Data Preprocessing

As part of data preprocessing, we removed potentially noisy sentence pairs using the heuristics presented in Data Filtering subsection. To handle rare words and out of vocabulary words in the corpus we tokenized the training data using sentencepiece Kudo and Richardson (2018).

4.1.1 Data Filtering

As most of the corpora is extracted by automated techniques, there are chances of presence of noisy sentence pairs in the corpus. As transformer is known to be sensitive to corpus noise Liu et al. (2019) rigorous filtering was performed on the corpus based on heuristics adopted from Li et al. (2019), Vegi et al. (2021) and Pinnis (2018). Details of the heuristics used are listed below.

- F0: Filter out sentence pair, in which either source or target sentence is empty.
- F1: Filter out sentence pair, in which either source or target sentence length greater than 800 characters.
- F2: Filter out sentence pair in which length of source and target sentence ratio is greater than 2.5.
- F3: Filter out sentence pair in which length of source and target sentence ratio is less than 0.4.
- F4: Filter out sentence pair, if source or target sentence contains word having length greater than 10.
- F5: Filter out sentence pair, if source and target sentences are equal.
- F6: Filter out sentence pair, if source or target sentence length is less than 4.

Corpus statistics after applying heuristics based filtering is given in Table 1. By applying heuristics, approximately 31% of total parallel sentences amounting to 44802801 are removed as they are potentially noisy pairs. Relative impact of each filter is also captured in Table 1. Heuristics chosen are language agnostic but there is always a room for corpus and language dependent heuristics, specifically the threshold values.

Experiments are carried out to observe the effect of data filtering (Configuration B vs Configuration A). Configuration A and B are discussed in detail in section 5.

[2] https://statmt.org/wmt22/large-scale-multilingual-translation-task.html

[3] https://github.com/pavanpankaj/Web-Crawl-African

Table 1 Statistics of training data before and after applying heuristic based filtering
%Filt is wrt previous filter and cumm %Filt is wrt Raw corpus

African↔English	Raw	F1-filt	%Filt	F1 +F2+F3	%Filt	F1+F2+F3+F4	%Filt	F1+F2+F3+F4+F5	%Filt	F1+F2+F3+F4+F5+F6	%Filt	cumm %Filt
afr-en	14357809	14331047	0.19%	14258195	0.005%	13675966	4.08%	13586470	0.65%	12128497	10.73%	15.5%
amh-en	1192934	1192625	0.026%	1142002	4.24%	1128660	1.16%	1115194	1.19%	946778	15.10%	20.6%
nya-en	1548650	1548650	0%	1529186	1.25%	1519738	0.61%	1519738	0%	1415637	6.84	8.5%
hau-en	9114633	9113895	0.008%	4164957	54.30%	3729275	10.46%	3712130	0.46%	3349586	9.76%	63.2%
ibo-en	519236	517737	0.29%	500379	3.35%	492926	1.49%	473208	4.0%	372787	21.22%	28.2%
kam-en	1656152	1656152	0%	1617111	2.35%	1616089	0.06%	1616088	6.18%	1452332	10.13%	12.3%
kin-en	9881964	9880973	0.010%	9715917	1.67%	9603289	1.15%	9603287	2.08%	8595328	10.49%	13.02%
lin-en	2890688	2890688	0%	2833279	1.98%	2826725	0.23%	2826725	0	2294855	18.81%	20.6%
lug-en	3478641	3476981	0.004%	3399032	2.24%	3356346	1.26%	3356345	1*%	2667772	20.51%	23.3%
luo-en	2767133	2767133	0%	2724060	1.55%	2719714	0.16%	2719714	0%	2339916	14.0%	15.4%
fuv-en	1376106	1376105	0%	1356236	1.44%	1349177	0.52%	1349172	0.0003%	1256816	6.84%	8.6%
nso-en	3087818	3087812	0%	3014807	2.36%	3009047	0.19%	3004799	0.14%	2284885	23.96%	26.00%
orm-en	2793892	2793892	0%	2738209	1.99%	2703241	1.28%	2703241	0%	2139879	20.84%	23.4%
sna-en	8933636	8933542	0%	8709596	2.51%	8625135	0.97%	8625118	0.0001%	7335877	14.95%	17.88%
som-en	1459349	1458307	0.0007%	1358266	6.86%	1336338	1.61%	1321903	1.08%	1084345	17.97%	25.6%
swh-en	32811268	32805580	0.0001%	32374856	0.013%	32154373	0.68%	32022095	0.4%	28152884	12.08%	14.2%
ssw-en	165712	165712	0%	154561	6.73%	152334	1.44%	152334	0	93832	38.40%	43.3%
tsn-en	5931529	5931529	0%	5667299	4.45%	5614356	0.93%	5614356	0	4257859	24.16%	28.2%
umb-en	302951	302951	0%	295177	2.57%	294655	0.18%	294654	0.0003%	247063	16.15%	18.44%
wol-en	208084	208073	13.09*%	204758	1.59*%	202100	1.3%	201928	0.08%	138994	31.17%	33.2%
xho-en	29326727	29326373	0%	9926807	66.15%	9795968	1.31%	9775666	0.20%	7552496	22.74%	74.24%
tso-en	638447	638382	0.0001%	620738	2.76%	619539	0.19%	619480	0.009%	511184	17.48%	19.9%
yor-en	1710752	1709669	0.0006%	1665254	2.59%	1651573	0.82%	1630170	1.29%	1471404	9.74%	13.9%
zul-en	4091851	4091355	0.0001%	3969983	2.97%	3928045	1.06%	3917179	0.28%	3352155	14.42%	18.1%
Total	140245962	140205163	0.002%	113940665	18.7%	112104609	1.6%	111760994	0.3%	95443161	14.6%	31.2%

4.1.2 Tagging of Source Sentences

As most of the African languages follow Latin script, so as to tag input sentences based on languages we have added special tokens at the source side similar to Vegi et al. (2021). Tokens are generated using special symbols of length 4. Special symbols are used to avoid overlapping of tags with language vocabularies.

4.2 Vocabulary Selection

We experimented with various configurations of source side sentencepiece subword vocabularies. However, for target side we fixed sentence piece subword vocabulary size to 16K for all the configurations.

Source side vocabulary estimation is done based on the work of Gowda and May (2020), where it is shown that for low resource languages optimal BLEU score is obtained for relatively smaller subword vocabulary of size between 4K to 6K. Also as most of the African languages follow Latin script, there are also chance of large vocabulary(subword) overlap among the languages.

1. Source side vocabulary is set to 100K, jointly for all 24 languages and used in Configurations A,B,C and D. Please refer to Section 5 for more details on Configurations.

2. Language wise 4K to 6K subword vocabulary based on language corpus size, where 6K is used for the languages having more than 1 million sentence pairs and 4K for languages having less than 1 million size. Though it is expected a total vocabulary of around 130K but we obtained 75K combined vocabulary as there are many common subword vocabulary among languages. This is used in Configuration E.

3. We experimented with increasing source side joint vocabulary from 100K to 144K in which 120K subword vocabulary for top 18 high resource languages and remaining 24K for the remaining 6 languages.

4.3 Model Training

ANVITA African MT system used base transformer, deep transformer, ensemble techniques and used fairseq framework for training Ott et al. (2019).

4.3.1 Base Transformer: 6x6

Training configuration follows base transformer similar to Vaswani et al. (2017) and used 6 encoder and 6 decoder layers. Base transformer model is trained on all corpora provided by the organizer except WebCrawl African corpora.

4.3.2 Deep Transformer: 24x6

Training used 24 encoder and 6 decoder layers for 10 epochs with batch size 10240, dropout 0.3, word embedding size of 1024, adam optimizer, update-freq 8, heads 8, encoder and decoder feed forward dimension of 4096, batch type tokens, warm-up steps 4000, learning rate $5e^{-4}$. Training configurations are adopted from Yang et al. (2021). Constrained and Primary models are trained on all corpora provided by Organizers except WebCrawl

African corpora.

4.3.3 Ensemble

We ensembled last two epochs of Deep Transformer 24x6 i.e 11,12 and this was our primary submission for the shared task.

5 Experimental Evaluation and Result Analysis

Experiments carried out for 6 distinct configurations to assess effect of filtering, deep transformer and strategies used for vocabulary selection.

5.1 Configurations

- Configuration A: Experiment is carried out for 10 epochs on Base Transformer architecture with 6 encoder and 6 decoder layers with out applying data filtering on all corpus provided by WMT 22 except WebCrawl African corpora.

- Configuration B: Experiment is carried out for 10 epochs on Base Transformer architecture with 6 encoder and 6 decoder layers with heuristic based filtering on all corpus provided by WMT 22 except WebCrawl African corpora.

- Configuration C: Experiment is carried out for 10 epochs on Deep Transformer architecture (as discussed in 4.2.2) with 24 encoder and 6 decoder layers with heuristic based filtering on all corpus provided by WMT 22 except WebCrawl African corpora.

- Configuration D: Experiment is carried out for 10 epochs on Deep Transformer architecture (as discussed in 4.2.2) with 24 encoder and 6 decoder layers with heuristic based filtering on all corpus provided by WMT 22 including WebCrawl African corpora.

- Configuration E: Experiment is carried out for 10 epochs on Deep Transformer architecture (as discussed in 4.2.2) with 24 encoder and 6 decoder layers with heuristic based filtering and language wise subword vocabulary(as discussed in 4.1.2) on all corpus provided by WMT 22 including WebCrawl African corpora.

- Configuration F: Configuration C is carried out for 2 more epochs (i.e. 11 and 12) and applied ensembling of last 2 epochs i.e. 11 and 12.

5.2 Results and Analysis

ANVITA African→English MT system was evaluated on standard Flores200 dataset Costa-jussà et al. (2022) and evaluation was also done by the organizer of Large-Scale Machine Translation Evaluation for African Languages task on blind test sets Adelani et al. (2022b). Results of both the experiments are given below Tables 2 and 3. Configuration-F is our primary submission and Configuration-C is our Contrastive submission to the WMT 2022 shared task on Large-Scale Machine Translation Evaluation for African Languages. Due to computational and time constraints we were not able to submit a model with WebCrawl African corpora as a primary/constrained submission. All the experiments carried out on Nvidia RTX 8000 48GB single GPU system. Training base transformer (6×6) without filtering and with filtering took approximately 400 hours and 80 hours respectively for 10 epochs. Remaining all experiments used deep transformer took around 290 hours for 10 epochs.

Table 2 shows the results obtained when experiments are carried out with configurations A,B,C,D,E, and F.

In the following subsections, key insights obtained using configurations A,B,C,D, and E are presented with respect to effect of filtering, deep transformer, and individual language wise subword vocabulary selections. However Configuration F is not compared against other configurations, as Configuration F is a replica of Configuration C with 12 epochs and did not use WebCrawl African corpora.

5.2.1 Effect of Filtering: Configuration A vs B

1. Heuristic based filtering has shown significant improvement on BLEU and CHRF2++ ranging from 0.04-2.06 and 0.23-1.55 respectively on all 22 out of 24 African → English language directions.

2. Reduced training time from 400 hours (Configuration A) to 80 hours (Configuration B).

3. Decrease in BLEU score and CHRF2++ for two languages namely Nigerian Fulfulde(fuv) and Wolof(wol).

Table 2 Results of African to English models on Flores200
Tran: Transformer, (n): refers to n epochs, prim: Primary model submitted to task,
Contras: Contrastive model submitted to task, ISV:Individual subword vocabulary,
WA:WebCrawl African(corpus submitted as part of the task)

	A: Tran 6 × 6 (10)		B :Tran 6 × 6 + filt(10)		C:Tran 24 × 6 + filt (10)(Contras)		D:Tran 24 × 6 + filt + WA(10)		E:Tran 24 × 6 + filt + WA + ISV (10)		F:Tran 24 × 6 + filt + ensem (Prim) (11,12)	
Afr↔En	BLEU	CHRF2++	BLEU	CHRF2++	BLEU	CHRF2++	BLEU	CHRF2++	BLEU	CHRF2++	BLEU	CHRF2++
afr	50.97	70.64	51.8	71.28	**55.8**	74.185	55.73	**74.21**	55.56	74.07	56.38	74.52
amh	16.45	40.01	17.29	41.14	**24.39**	48.80	24.17	**48.82**	24.45	48.98	24.78	49.46
nya	18.42	40.09	18.5	40.90	22.45	**48.79**	**22.66**	45.46	22.35	44.37	22.90	45.148
hau	21.42	43.74	22.3	45.09	27.92	49.95	28.04	**50.18**	**28.25**	50.06	28.97	50.70
ibo	15.48	37.15	15.9	38.81	20.62	44.07	21.25	**44.44**	**22.35**	44.37	21.79	44.93
kam	6.98	23.86	7.44	24.65	9.24	28.26	**9.49**	**28.33**	8.78	27.09	9.41	27.91
kin	19.90	42.38	21.7	43.5	25.97	48.01	**26.15**	**48.34**	25.48	47.38	25.83	48.11
lin	14.26	35.06	15.22	36.34	19.34	40.80	**19.56**	**41.2**	18.18	39.82	19.4	40.82
lug	12.10	31.99	13.13	33.37	15.93	37.09	**16.69**	**37.73**	16.44	37.48	16.30	37.37
luo	13.41	33.64	13.08	33.87	17.34	**38.51**	**16.96**	38.32	16.62	38.04	17.54	38.58
nso	23.68	44.71	25.6	46.96	33.30	53.77	**33.54**	**54.52**	33.22	53.96	34.02	54.36
fuv	5.13	20.99	4.5	19.72	5.62	21.91	**5.82**	**21.95**	5.12	19.95	5.71	21.54
orm	6.75	24.70	7.38	26.24	11.27	31.55	**12.13**	**33.57**	11.94	33.06	11.67	32.05
sna	19.94	42.68	19.98	42.98	23.68	46.429	23.57	**46.73**	**24.23**	46.17	24.25	46.61
som	13.75	34.76	13.96	35.01	18.01	40.02	**17.80**	**40.02**	17.37	39.55	18.07	40.22
swh	33.71	56.30	35.77	57.85	41.01	62.23	**41.19**	**62.32**	40.60	61.99	41.34	62.49
ssw	16.73	38.00	17.61	39.18	23.68	45.79	**25.34**	**47.27**	24.6	46.61	24.49	46.15
tsn	18.01	39.7	18.35	40.63	22.66	44.97	**23.08**	**45.96**	22.88	45.27	23.2	45.65
tso	19.02	39.80	19.38	40.64	24.32	45.85	21.72	44.33	**25.35**	**47.09**	24.5	46.04
umb	3.98	20.58	4.33	21.57	**5.74**	**24.35**	5.55	24.27	5.41	23.44	5.65	23.87
wol	5.64	23.02	4.93	21.75	8.71	27.10	8.43	27.01	**8.85**	**27.35**	8.71	27.17
xho	25.01	47.1	25.18	47.49	31.8	53.78	32.01	53.84	**33.47**	**54.97**	32.53	54.09
yor	11.01	31.14	12.20	32.54	15.3	37.12	15.39	37.20	**15.98**	**38.14**	15.58	37.45
zul	27.07	49.64	28.17	51.01	33.4	55.52	33.79	55.70	**34.68**	**56.06**	34.34	55.78

5.2.2 Effect of Deep Transformer: Configuration B vs C

1. Deep transformer architecture (Configuration C) has shown significant improvement on BLEU and CHRF2++ ranging 1.12-7.7 and 2.19-7.89 respectively on all 24 African → English language directions.

2. As expected, it increased training time from 80 hours (Configuration B) to 290 hours (Configuration C), but still less than base transformer training time without filtering.

5.2.3 Effect of inclusion of WebCrawl African: Configuration C vs D

1. Inclusion of Our corpora-3, WebCrawl African (Configuration D) has shown improvement on BLEU ranging 0.01-1.66 for 12 out of 15 African→English translation directions and even by +0.18-0.68 for the 4 out of 9 African→English translation directions. However there is a marginal decrease in remaining African→English directions.

2. Inclusion of Our corpora-3, WebCrawl African (Configuration D) has shown improvement on CHRF2++ ranging 0-1.48 on 19 African → English language directions, however there is a marginal decrease in remaining directions.

5.2.4 Effect of ISV (Individual Subword Vocabulary): Configuration D vs E

ISV (Configuration E) has shown significant improvement on few language directions, however there is a marginal decrease of BLEU and CHRF2++ in majority of the directions and specifically 17 and 19 out of 24 respectively.

It is observed that increase of source side joint vocabulary beyond 100K does not improve performance and in fact decrease in BLEU score is observed for majority of the languages. Also use of language wise vocabulary selection did not consistently improve performance of low resource languages in comparison to joint vocabulary selection.

5.3 Comparison With Available Models

To the best of our knowledge, results using a single multilingual model covering all the 24 African languages to English is not available. Often meaningful comparison becomes hard as not all the reported results use same test-set used here. Yang et al. (2021) trained NMT model for translating

Table 3 Results of African to English models on blind test set from Organizer
Tran: Transformer, (n): refers to n epochs, blind:refers to blind test set used by Organizer for evaluation, prim: primary model submitted to the task, Contras: Contrastive model submitted to the task,
* represents the languages where evaluation was not provided by the Organizer

Afr↔En	F: Tran 24 × 6 (11,12)+ensemble(Prim)			C:Tran 24 × 6 (10) (Contras)		
	BLEU	spBLEU	CHRF2++	BLEU	spBLEU	CHRF2++
afr	56.1	59	74.4	55.8	58.7	74.2
amh	24.8	26	48.5	24.1	25.2	47.8
nya	23.8	26.5	45.7	23.1	26.2	45.5
hau	30.3	32.6	51.7	28.8	31.3	50.9
ibo	24.8	27.1	47.2	23.6	25.8	46.2
kam	10.3	12.4	28.2	10.3	12.4	28.4
kin	27.7	29.2	48.9	27.4	28.9	48.8
lin*	-	-	-	-	-	-
lug	16.6	18.7	37.2	16.5	18.5	36.7
luo	17.9	19.9	38.3	17.6	19.5	37.9
fuv	6.2	8	21.9	6.1	7.9	22
nso	34.1	35.9	54.1	33.7	35.5	53.6
orm	11.9	12.6	31.8	11.2	12	31.5
sna	25.3	28	46.7	24.6	27.6	46.3
som	21	22.7	42	20.7	22.2	41.4
swh	40.6	42	61.3	40.4	41.7	61
ssw	25.9	27.9	46.7	25.5	27.5	46.2
tsn	26.2	28.2	47.7	25.4	27.5	47.1
umb	6.4	8.2	24.6	6.2	8.1	24.7
wol*	-	-	-	-	-	-
xho	30	32.4	51.6	29.8	32.4	51.6
tso	25.3	27.4	46.2	25.3	27.2	46
yor	16.3	18.4	37.5	15.8	17.9	37
zul	33.6	35.6	54.4	32.5	34.9	54

101 languages from any to any directions and 12 out of 24 translation directions part of our submission are in common. Comparison on FLORES shows our model produced an improved results for 7 out of 12 African→English directions namely {Hausa, Chichewa, Swahili, Xhosa, Yoruba, Zulu }→English. Emezue and Dossou (2021) trained many to many models for African languages and 5 out 24 African translation directions part of our submission are in common. Our model showed an improvement for all the common 5 African→English directions namely {Igbo, Kinyarwanda, Xhosa, Yoruba, Swahili}→English.

6 Conclusion

This paper describes our submission to WMT 2022 shared task on Large-Scale Machine Translation Evaluation for African Languages under the constrained translation track. We focused on 24 African languages to English MT directions. Multilingual model with deep transformer showed significant improvement in BLEU and CHRF2++ scores across all 24 African to English MT directions. Vocabulary size of 4K to 6K per language for estimating size of joint source vocabulary seems to be a good choice in a multilingual setup. Heuristic based filtering did improve the BLEU scores. However the biggest gain of filtering observed is in terms of training time speed up by 5x. Empirical results indicate that training using deep transformer with filtered corpora seems to be a better choice than using base transformer on the whole corpora both in terms of MT accuracy and training time.

Acknowledgements

The authors would like to thank Director, CAIR for his constant encouragement, guidance and enablement.

References

David Adelani, Jesujoba Alabi, Angela Fan, Julia Kreutzer, Xiaoyu Shen, Machel Reid, Dana Ruiter, Dietrich Klakow, Peter Nabende, Ernie Chang, Tajuddeen Gwadabe, Freshia Sackey, Bonaventure F. P. Dossou, Chris Emezue, Colin Leong, Michael Beukman, Shamsuddeen Muhammad, Guyo Jarso, Oreen Yousuf, Andre Niyongabo Rubungo, Gilles Hacheme, Eric Peter Wairagala, Muhammad Umair Nasir, Benjamin Ajibade, Tunde Ajayi, Yvonne Gitau, Jade Abbott, Mohamed Ahmed, Millicent Ochieng, Anuoluwapo Aremu, Perez Ogayo, Jonathan Mukiibi, Fatoumata Ouoba Kabore, Godson Kalipe, Derguene Mbaye, Allahsera Auguste Tapo, Victoire Memdjokam Koagne, Edwin Munkoh-Buabeng, Valencia Wagner, Idris Abdulmumin, Ayodele Awokoya, Happy Buzaaba, Blessing Sibanda, Andiswa Bukula, and Sam Manthalu. 2022a. A few thousand translations go a long way! leveraging pre-trained models for African news translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3053–3070, Seattle, United States. Association for Computational Linguistics.

David Ifeoluwa Adelani, Md Mahfuz Ibn Alam, Antonios Anastasopoulos, Akshita Bhagia, Marta Costa-Jussá, Jesse Dodge, Fahim Faisal, Christian Federmann, Natalia Fedorova, Francisco Guzmán, Sergey Koshelev, Jean Maillard, Vukosi Marivate, Jonathan Mbuya, Safiyyah Saleem, and Holger Schwenk. 2022b. Findings of the WMT 2022 shared task on large-scale machine translation evaluation for african languages. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Chris Chinenye Emezue and Bonaventure F. P. Dossou. 2021. MMTAfrica: Multilingual machine translation for African languages. In *Proceedings of the Sixth Conference on Machine Translation*, pages 398–411, Online. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(107):1–48.

Thamme Gowda and Jonathan May. 2020. Finding the optimal vocabulary size for neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3955–3964, Online. Association for Computational Linguistics.

Thamme Gowda, Zhao Zhang, Chris Mattmann, and Jonathan May. 2021. Many-to-English machine translation tools, data, and pretrained models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 306–316, Online. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual NMT representations at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1565–1575, Hong Kong, China. Association for Computational Linguistics.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, et al. 2019. The niutrans machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266.

Hairong Liu, Mingbo Ma, Liang Huang, Hao Xiong, and Zhongjun He. 2019. Robust neural machine translation with joint textual and phonetic embedding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3044–3049, Florence, Italy. Association for Computational Linguistics.

Owen McOnyango, Florence Indede, Lilian D.A. Wanzare, Barack Wanjawa, Edward Ombui, and

Lawrence Muchemi. 2022. Kencorpus: Kenyan Languages Corpus.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Mārcis Pinnis. 2018. Tilde's parallel corpus filtering methods for wmt 2018. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 939–945.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Pavanpankaj Vegi, J Sivabhavani, Biswajit Paul, Chitra Viswanathan, and Prasanna Kumar KR. 2021. Anvita machine translation system for wat 2021 multiindicmt shared task. In *Proceedings of the 8th Workshop on Asian Translation (WAT2021)*, pages 244–249.

Lilian D.A Wanzare, Florence Indede, Owen McOnyango, Edward Ombui, Barack Wanjawa, and Lawrence Muchemi. 2022. KenTrans: A Parallel Corpora for Swahili and local Kenyan Languages.

Jian Yang, Shuming Ma, Haoyang Huang, Dongdong Zhang, Li Dong, Shaohan Huang, Alexandre Muzio, Saksham Singhal, Hany Hassan, Xia Song, and Furu Wei. 2021. Multilingual machine translation systems from Microsoft for WMT21 shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 446–455, Online. Association for Computational Linguistics.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020a. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Yuhao Zhang, Ziyang Wang, Runzhe Cao, Binghao Wei, Weiqiao Shan, Shuhan Zhou, Abudurexiti Reheman, Tao Zhou, Xin Zeng, Laohu Wang, et al. 2020b. The niutrans machine translation systems for wmt20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 338–345.

Shuhan Zhou, Tao Zhou, Binghao Wei, Yingfeng Luo, Yongyu Mu, Zefan Zhou, Chenglong Wang, Xuanjun Zhou, Chuanhao Lv, Yi Jing, Laohu Wang, Jingnan Zhang, Canan Huang, Zhongxiang Yan, Chi Hu, Bei Li, Tong Xiao, and Jingbo Zhu. 2021. The NiuTrans machine translation systems for WMT21. In *Proceedings of the Sixth Conference on Machine Translation*, pages 265–272, Online. Association for Computational Linguistics.

HW-TSC Systems for WMT22 Very Low Resource Supervised MT Task

Shaojun Li, Yuanchang Luo, Daimeng Wei, Zongyao Li, Hengchao Shang,
Xiaoyu Chen, Zhanglin Wu, Jinlong Yang, Zhiqiang Rao, Zhengzhe Yu,
Yuhao Xie, Lizhi Lei, Hao Yang, Ying Qin
Huawei Translation Service Center, Beijing, China
{lishaojun18,luoyuanchang,weidaimeng,lizongyao,shanghengchao,chenxiaoyu35,
wuzhanglin2,yangjinlong7,raozhiqiang,yuzhengzhe,xieyuhao2,
leilizhi,yanghao30,qinying}@huawei.com

Abstract

This paper describes the submissions of Huawei translation services center (HW-TSC) to the WMT22 Very Low Resource Supervised MT task. We participate in all 6 supervised tracks including all combinations between Upper/Lower Sorbian (Hsb/Dsb) and German (De). Our systems are build on deep Transformer with a large filter size. We use multilingual transfer with German-Czech (De-Cs) and German-Polish (De-Pl) parallel data. We also utilize regularized dropout (R-Drop), back translation, fine-tuning and ensemble to improve the system performance. According to the official evaluation results on OCELoT[1], our supervised systems for all 6 language directions got the highest BLEU scores among all submissions. Our pre-trained multilingual model for unsupervised De2Dsb and Dsb2De translation also gains the highest BLEU.

1 Introduction

In this paper, we describe our very low resource supervised MT systems for all combinations between Hsb, Dsb and De. We first select a base pre-trained multilingual model and then fine-tune it. As we focus primarily on the supervised task, we only apply our pre-trained multilingual system with zero-shot for unsupervised task submissions.

As show in WMT21 shared task (Libovický and Fraser, 2021), most participants use De-Cs for transfer or combine De-Cs with the low resource pairs to build a multilingual system. Fine-tuning based on a multilingual pre-trained (Fan et al., 2020) model has shown very promising results for low resource tasks. We add De-Pl data and train our multilingual pre-trained model to transfer the low resource pairs.

This paper is structured as follows: we describe our data source and data pre-processing method in section 2. We detail the model structure and method we used in Section 3. We then present the final experiments in Section 4 and Section 5, and finally we conclude our work in Section 6.

2 Dataset

2.1 Data Source

For our base pre-trained multilingual systems, we use all the bilingual data (De-Cs and De-Pl) from the latest version of OPUS. We also sample 20M German monolingual data from news (general) MT task for augmentation. For fine-tuning the systems transfer to our task, we use all the bilingual and monolingual data officially provided without any filtering strategy. We use dev set and test set together for model parameter adjustment and system selection (do not include the blind test data from the previous years).

2.2 Data Pre-processing

For all the data mentioned above, we remove duplicate sentences (Khayrallah and Koehn, 2018; Ott et al., 2018).

For De-Cs and De-Pl, the data pre-processing procedure is as follows:

- Remove sentences with mismatched parentheses and quotation marks.
- Filter out sentences of which punctuation percentage exceeds 0.4.
- Filter out sentences with a character-to-word ratio greater than 12 or less than 1.5.
- Filter out sentences with more than 150 words.
- Apply langid (Joulin et al., 2017, 2016) to filter out sentences in other languages.
- Use fast-align (Dyer et al., 2013) to filter out sentence pairs that are poorly aligned.

[1] https://ocelot-wmt22.mteval.org

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1098 - 1103
December 7-8, 2022 ©2022 Association for Computational Linguistics

	bilingual					**monolingual**		
	De-Cs	**De-Pl**	**De-Dsb**	**De-Hsb**	**Dsb-Hsb**	**De**	**Dsb**	**Hsb**
Raw data	77.1M	98.1M	40K	449K	63K	-	220K	1.13M
Processed data	55.9M	66.5M	39K	317K	63K	20M	177K	957K

Table 1: The data sizes of before and after pre-processing in our very low resource supervise MT Task

We sample 3.2M Cs and Pl data from bilingual data and up-sampling 3.2M De, Hsb, Dsb from a combined dataset of all the three languages. We mix the data above and build a joint SentencePiece model (SPM) (Kudo and Richardson, 2018; Kudo, 2018) for word segmentation, with a vocabulary of 40k. We use Moses tokenizer (Koehn et al., 2007) to pre-segment sentences. We also use the combined data to build a joint vocabulary for all of our models. The vocabulary size is slightly larger than SPM vocabulary to cover more tokens, which is set to 41k.

3 System Overview

3.1 Model

Transformer (Vaswani et al., 2017), as the current mainstream architecture of NMT, adopts a fully self-attention mechanism, which can realize algorithm parallelism, speed up model training, and improve model performance. Deep transformer is an variant of Transformer, which increases the number of encoder layers and uses pre-layer-normalization to further improve model performance. Therefore, in all translation tasks, we adopt the following model architecture:

- Deep Transformer (Wei et al., 2021): We refer to the Transformer-big model architecture and decrease the dim for faster training. our Deep Transformer model features pre-layer-normalization, 35-layer encoder, 6-layer decoder, 16-head self-attention, 768-dimension word embedding and 3072-hidden-state.

3.2 Multilingual Transfer

Recent researches have shown that multilingual models outperform their bilingual counterparts, particularly when the number of languages in the system is limited and those languages are related (Lakew et al., 2018). This is mainly due to the capability of the model to learn interlingual knowledge (shared semantic representation between languages) (Johnson et al., 2016) (Ranathunga et al., 2021). Transfer learning using pre-trained multilingual model (Fan et al., 2020) has shown very promising results for low resource tasks. In this task, we first select a multilingual system as the base system, then fine-tune the system with low resource language pairs.

3.3 R-Drop

Dropout (Srivastava et al., 2014) is a powerful and widely used technique for regularizing deep neural networks. Though it can help improve training effectiveness, the randomness introduced by dropouts may lead to inconsistencies between training and inference. R-Drop (Wu et al., 2021) forces the output distributions of different sub models generated by dropout be consistent with each other. Therefore, we use R-Drop to augment the pre-trained multilingual model for each track and reduce inconsistencies between training and inference.

3.4 Back Translation

Back translation (BT) (Edunov et al., 2018) refers to translating the target monolingual data into the source language, and then using the synthetic data to increase the training data size. This method has been proven effective to improve the NMT model performance. We apply sampling(Graça et al., 2019) back-translation for all language directions.

4 Experimental Settings

During the training phase, we use Pytorch-based Fairseq[2] (Ott et al., 2019) open-source framework. Each model is trained using 8-V100 with a batch size of 2048 tokens for each GPU. Dropout was set to 0.1 for pre-train multilingual model, and 0.3 for fine-tuning model. The label smoothing rate (Szegedy et al., 2016) is 0.1. Adam optimizer (Kingma and Ba, 2015) with $\beta 1$=0.9 and $\beta 2$=0.98 is also used. Furthermore, we use reg_label_smoothed_cross_entropy as the loss function and set reg-alpha to 5 when applying R-

[2] https://github.com/facebookresearch/fairseq

System	De2Hsb	Hsb2De
Pre-trained model	1.2	3.6
Bitext finetune	65.6	66.3
Noisied ST	65.8	-
Sampling BT	69.2	67.0
FT+ST	-	67.1
Ensemble	69.4	67.5
WMT22submission	**70.7**	**71.9**

Table 2: Avg. scores on WMT21 dev set, test set and WMT22 dev set for De↔Hsb.

System	De2Dsb	Dsb2De
Pre-trained model	0.9	2.5
Bitext finetune	50.1	55.2
Noisied ST	50.6	-
Sampling BT	58.0	57.8
FT+ST	-	57.9
Ensemble	58.2	58.1
WMT22submission	**73.9**	**62.5**

Table 3: Avg. scores on WMT21 dev set, test set and WMT22 dev set for De↔Dsb.

Drop training strategy. For pre-training the multilingual model, the update frequency, the learning rate and warm-up steps are 4, 5e-4 and 4000 respectively; for fine-tuning the model, the update frequency and the learning rate is 1 and 1e-4 without warm-up. In the evaluation phase, we use Marian[3] (Junczys-Dowmunt et al., 2018) for decoding and then calculate the sacreBLEU[4] (Post, 2018) on the WMT21 dev set, test set and WMT22 dev test to measure the performance of each model.

5 Experimental Result

First of all, we test pre-trained multilingual models without fine-tune and get quiet low scores. Next, we fine-tune all of our pre-trained models with bitext and then select a best one according to the BLEU scores for every task.

5.1 De↔Hsb

Table 2 shows the results of using the selected pre-trained multilingual model to improve the De↔Hsb model performance.

System	Hsb2Dsb	Dsb2Hsb
Pre-trained model	1.1	1.0
Bitext finetune	62.9	65.3
Multilingual finetune	67.6	72.0
Sampling BT	69.6	74.2
Ensemble	70.0	74.7
WMT22submission	**88.0**	**86.8**

Table 4: Avg. scores on WMT22 dev set for Dsb↔Hsb.

In De2Hsb, we adopt the strategy of noised ST (Imamura and Sumita, 2018) because we have a large amount of German monolingual data. We sample 20M German monolingual for noised ST. We find that this strategy can bring an additional 0.2 BLEU improvement. At the same time, we use all the Hsb monolingual data (including the Hsb side of Dsb-Hsb) for sampling BT, which brings an increase of 3.4 BLEU. BLEU increases by 0.2 after ensemble.

In Hsb2De, We find that both sampling BT and forward translation + sampling BT (FT+ST) (Wu et al., 2019) can bring certain improvement, but sampling BT outperforms the FT+ST strategy (0.7 BLEU vs 0.1 BLEU). After ensemble, model performance continues improving by 0.4 BLEU.

5.2 De↔Dsb

Table 3 shows the results of using the selected pre-trained multilingual model to improve the De↔Dsb model performance. We follow the same strategy as that of De↔Hsb.

In De2Dsb, we adopt noised ST with the same data as De2Hsb. We use 20M German monolinguals for noised ST. We find that this strategy can bring an additional 0.5 BLEU improvement. At the same time, we use all the Dsb monolingual data (including the Dsb side of Dsb-Hsb) for sampling BT, which brings an improvement of 7.4 BLEU. Finally, BLEU increases by 0.2 after ensemble.

In Dsb2De, Sampling BT and FT+ST can bring certain improvement (2.6 BLEU vs 0.1 BLEU). After ensemble, model performance continues improving by 0.2 BLEU.

5.3 Hsb↔Dsb

Table 4 shows the results of using the selected pre-trained multilingual model to improve the Hsb↔Dsb model performance.

Regarding the Hsb2Dsb task, we first fine-tune the many-to-many pre-trained model with bitext, and then combine the De2Hsb, De2Dsb, Hsb2De, Dsb2De multilingual to continue fine-tuning both Hsb2Dsb and Dsb2Hsb models. This strategy gets improvements of 5+ BLEU. Then, we perform one

[3] https://github.com/marian-nmt/marian

[4] https://github.com/mjpost/sacreBleu

Pre-trained model	De2Hsb	De2Dsb	Hsb2De	Dsb2De	Hsb2Dsb	Dsb2Hsb
De2Cs	64.5	49.1	-	-	61.0	63.2
Cs2De	-	-	64.7	51.9	-	-
one-to-many	**64.6**	**49.4**	-	-	61.8	63.6
many-to-one	-	-	**65.3**	**54.0**	-	-
many-to-many	64.6	49.2	64.9	53.0	**62.0**	**64.3**

Table 5: Avg. scores on WMT21 dev set,test set and WMT22 dev set with different pre-trained models. **De2Cs**: pre-train with De2Cs bilingual data; **Cs2De**: pre-train with Cs2De bilingual data; **one-to-many**: pre-train with De2Cs and De2Pl bilingual data; **many-to-one**: pre-train with Cs2De and Pl2De bilingual data; **many-to-many**: pre-train with Cs2De, De2Cs, Pl2De and De2Pl bilingual data.

System	De2Hsb	De2Dsb	Hsb2De	Dsb2De	Hsb2Dsb	Dsb2Hsb
w/o R-drop	64.6	49.4	65.3	54.0	62.0	64.3
w/ R-drop	**65.6**	**50.1**	**66.3**	**55.2**	**62.9**	**65.3**

Table 6: Avg. scores of WMT21 dev set, test set and WMT22 dev set for each track without or with R-drop.

round of sampling BT for optimization.

After we ensemble the latter two models, the model performances significantly increase by 7.1 and 9.4 BLEU respectively when comparing with the base many-to-many fine-tuning model, which also proves the advantages of the multilingual model in low-resource tasks.

5.4 Unsupervised Submission

We conduct an unsupervised experiment with our many-to-one and one-to-many pre-trained model. For Hsb2De and Dsb2De, we add tags to the Hsb/Dsb monolingual data and get the German result from many-to-one model for zero-shot. Then we fine-tune the best one-to-many model with the zero-shooting translations, and get the Hsb2De, Dsb2De models, which obtains 11.5 BLEU on the Hsb2De track and 13.5 BLEU on the Dsb2DE track. Based on the two base models, we continue conducting a round of BT with 2M German monolingual data, and then train the De2Hsb and De2Dsb models for submission. The BLEU scores are 10.4 (De2Hsb) and 9.0 (De2Dsb). As we have not invested much efforts in the unsupervised task, more experiments need to be done in the feature.

6 Analysis

6.1 Pre-trained Model

Due to the availability of large amount of De-Cs and De-Pl data and the similarities between Hsb/Dsb and Cs/Pl, we pre-train several multilingual models with different strategies, and then fine-tune with very low resource bilingual data. We choose the best strategy for every language direction for further fine-tuning. Specifically, we design three pre-trained multilingual models: a one-to-many model trained with the De2Cs and De2Pl bilingual data, a many-to-one model trained with the Cs2De and Pl2De bilingual data, and a many-to-many model trained with the Cs2De, De2Cs, Pl2De and De2Pl bilingual data. For each corpus we use a different tag to differentiate. Furthermore, we train a De-Cs model as done by last year's participants, in order to get better comparison results.

Table 5 shows that data selection for the pre-trained model is closely related to the task requirements. For tasks of translating other languages into De, the many-to-one model trained with Cs2De and Pl2De corpora performs the best. For tasks of De2Hsb/Dsb, the one-to-many model trained with De2Cs and De2Pl corpora works the best. In general, the many-to-many model is more suitable for the Hsb2Dsb and Dsb2Hsb task, because the many-to-many model is trained with Cs and Pl data at both the source and target sides. With large amount of Cs/Pl data for transfer, both the encoding and decoding layers can benefit transfer for the Hsb2Dsb and Dsb2Hsb embeddings. Multilingual pre-trained models have better performance than bilingual pre-trained models in all directions. Besides, the unsupervised results also show the transfer capability of our pre-trained multilingual models.

6.2 The Effect of R-drop

Considering the limited sources and the large size of the multilingual model, we try the R-drop strat-

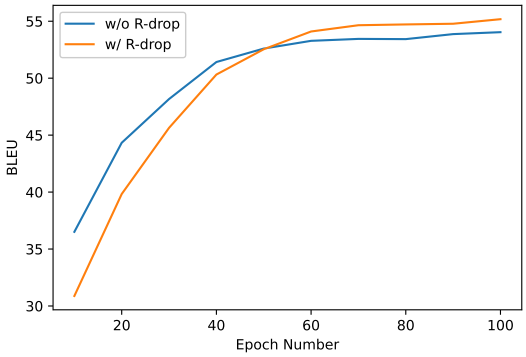


Figure 1: BLEU curves along with or without R-drop

egy to see whether the R-drop strategy is effective on multilingual models. Based on the optimal multilingual model for each task selected in the previous step, we compared whether the use of the R-drop strategy in the training process can lead to further improvements. It can be seen from Table 6 that after using the R-drop strategy for training, the BLEU of each track further improves by at least 1 point, indicating that R-drop does have a good effect in low-resource scenarios with large models. Therefore, we adopt the R-drop strategy for further training. In addition, Figure 1 is a graph depicting the BLEU convergence curves on the Dsb2De track. From the figure we can find that when the BLEU value increases by using R-drop, the convergence time also increases by about ten epochs.

7 Conclusion

This paper presents our translation systems for the WMT22 very low resource supervised MT task. During the experiment, We use multilinguals to build our base translation system, and then use forward translation and back translation methods to expand the size of training data for a better translation system. We also adopt test set fine-tuning and ensemble to further improve the system performance. Finally, according to the official evaluation results on OCELoT, our submission achieves the highest BLEU scores in all 6 language directions in the supervised task, and our submission of De2Dsb and Dsb2De also gains the highest BLEU in unsupervised task.

References

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *arXiv: Computation and Language*.

Miguel Graça, Yunsu Kim, Julian Schamper, Shahram Khadivi, and Hermann Ney. 2019. Generalizing back-translation in neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 45–52.

Kenji Imamura and Eiichiro Sumita. 2018. Nict self-training approach to neural machine translation at nmt-2018. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 110–115.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Herve Jegou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models.

Armand Joulin, Édouard Grave, Piotr Bojanowski, and Tomáš Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, et al. 2018. Marian: Fast neural machine translation in c++. In *ACL (4)*.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83.

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *ACL (1)*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP (Demonstration)*.

Surafel Melaku Lakew, Mauro Cettolo, and Marcello Federico. 2018. A comparison of transformer and recurrent neural networks on multilingual neural machine translation. *international conference on computational linguistics*.

Jindřich Libovický and Alexander Fraser. 2021. Findings of the WMT 2021 shared tasks in unsupervised MT and very low resource supervised MT. In *Proceedings of the Sixth Conference on Machine Translation*, pages 726–732, Online. Association for Computational Linguistics.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *ICML*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Surangika Ranathunga, En-Shiun Annie Lee, Marjana Prifti Skenduli, Ravi Shekhar, Mehreen Alam, and Rishemjit Kaur. 2021. Neural machine translation for low-resource languages: A survey. *arXiv: Computation and Language*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Daimeng Wei, Zongyao Li, Zhanglin Wu, Zhengzhe Yu, Xiaoyu Chen, Hengchao Shang, Jiaxin Guo, Minghan Wang, Lizhi Lei, Min Zhang, et al. 2021. Hwtsc's participation in the wmt 2021 news translation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 225–231.

Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. 2021. R-drop: Regularized dropout for neural networks. *Advances in Neural Information Processing Systems*, 34:10890–10905.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4207–4216.

PICT-NLP@WMT22-EMNLP2022: Unsupervised and Very-Low Resource Supervised Translation on German and Sorbian Variant Languages

Aditya Vyawahare [*]
aditya.vyawahare07@gmail.com

Rahul Tangsali [*]
rahuul2001@gmail.com

Aditya Mandke [†]
amandke@ucsd.edu

Onkar Litake [†]
olitake@ucsd.edu

Dipali Kadam [‡]
ddkadam@pict.edu

Pune Institute of Computer Technology, India

Abstract

This paper presents the work of team PICT-NLP for the shared task on unsupervised and very low-resource supervised machine translation, organized by the Workshop on Machine Translation, a workshop in collocation with the Conference on Empirical Methods in Natural Language Processing (EMNLP 2022). The paper delineates the approaches we implemented for supervised and unsupervised translation between the following 6 language pairs: German-Lower Sorbian (de-dsb), Lower Sorbian-German (dsb-de), Lower Sorbian-Upper Sorbian (dsb-hsb), Upper Sorbian-Lower Sorbian (hsb-dsb), German-Upper Sorbian (de-hsb), and Upper Sorbian-German (hsb-de). For supervised learning, we implemented the transformer architecture from scratch using the Fairseq library. Whereas for unsupervised learning, we implemented Facebook's XLM masked language modeling approach. We discuss the training details for the models we used, and the results obtained from our approaches. We used the BLEU and chrF metrics for evaluating the accuracies of the generated translations on our systems.

1 Introduction

Neural machine translation has witnessed significant progress in the case of highly spoken languages such as English (Bahdanau et al., 2015), Mandarin (Li et al., 2022), French (Emezue and Dossou, 2020), etc. However, in many cases, it becomes challenging to develop a robust bilingual machine translation system, especially with limited resources (Dong et al., 2015). There are big-tech companies such as Google[1] and Bing[2], which have taken initiatives to build translation systems for multiple languages. Still, languages that are low-resource in nature, such as the Sorbian family of languages (Howson, 2017), have gotten lesser attention in terms of research. The paper focuses on the development of machine translation systems between pairs of languages from German, Lower Sorbian, and Upper Sorbian, using both supervised and unsupervised approaches.

In the Indo-European language family, German (Deutsch) belongs to the western Germanic branch (Durrell, 2006). Approximately 95 million people speak it natively; 28 million speak it as a second language in more than 40 countries. Due to a phonetic mutation called High German Consonant Shift (Vennemann, 2008), German moved away from other Germanic languages. This shift in German consonants occurred between the 3rd and 5th centuries, and probably ended in the 9th century AD.

Lower Sorbian (dolnoserbska rěc) and Upper Sorbian (hornjoserbska rěč) are western Slavonic languages spoken in the region of Lower and Upper Lusatia in the southeast of Germany respectively. They are closely related to other West Slavonic languages, including Polish, Czech, Slovak, and Kashubian. There are seven recognized autochthonous minorities and regional languages in Germany, including Danish, Saterfrisian, North Frisian, Romanes, and Lower German.

We aimed to carry out research in neural machine translation between German, which is high-resource in nature, and Lower and Upper Sorbian, which are low-resource languages. We implement supervised and unsupervised methods for the same. For the supervised approach, we trained transformer (Vaswani et al., 2017) models from scratch using the bilingual data provided by WMT in the 2022 workshop edition. We used the Fairseq[3] (Ott et al., 2019) library for the same, which is a sequence-to-sequence learning toolkit for neural

[*] equal contribution

[†] equal contribution

[‡] equal contribution

[1] https://translate.google.com/

[2] https://www.bing.com/translator

[3] https://fairseq.readthedocs.io/en/latest/

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1104 - 1110
December 7-8, 2022 ©2022 Association for Computational Linguistics

Supervised	de-dsb	dsb-de	dsb-hsb	hsb-dsb	de-hsb	hsb-de
Parallel	40,194	40,194	62,565	62,565	70,000	70,000

Table 1: Statistics of the dataset used for supervised training

Unsupervised	de	dsb	hsb
Monolingual	53,309	1,45,198	2,22,027

Table 2: Statistics of the dataset used for unsupervised training

machine translation. Transformer(Vaswani et al., 2017) is a Seq2Seq (Sutskever et al., 2014a) model that uses self-attention to train on input data. The encoder part of the transformer model consists of a self-attention layer and a feed forward neural network (Bebis and Georgiopoulos, 1994). The encoder of the transformer reads the input sequence, one word at a time to produce a hidden vector. The decoder produces the output sequence from the vector received from the encoder. Being part of recent NMT research, transformers perform well compared to baseline models such as CNNs (Albawi et al., 2017) and LSTMs (Hochreiter and Schmidhuber, 1997).

For the unsupervised approach, implemented Facebook XLM's[4] masked language model (cross-lingual language model) for unsupervised learning (Chronopoulou et al., 2021). Training data used for the same was monolingual data provided by the organizers and the OPUS project[5]. We preprocessed the data, and also applied byte-pair encoding (BPE) (Sennrich et al., 2016) to the input and target data. We made use of the fastBPE[6] library for the same. Finally, we applied XLM preprocessing on the data before training.

We experimented our approaches on six language pairs between German, Lower Sorbian and Upper Sorbian. We have used the BLEU (Papineni et al., 2002) and chrF (Popović, 2015) evaluation metrics for computing accuracy, which have been discussed in this paper.

2 Dataset Description

We used the data provided by the WMT22 organizers, and from the OPUS project, recommended by the organizers. The statistics of the training data for both supervised and unsupervised approaches is given in Table 2. For the supervised training, we used the parallel data provided by the organizers, for each language pair. We used the 2022 version of the data itself, as training for larger corpora was proving computationally expensive at our end. For translations between German and Lower Sorbian, validation data size was 1353, whereas for Upper Sorbian-Lower Sorbian and German-Upper Sorbian, validation data sizes were 709 and 2000 for each language respectively.

For unsupervised learning, we used the monolingual data for Lower Sorbian provided by the organizers. For Upper Sorbian, we used the monolingual data provided by the Witaj Sprachzentrum[7]. Whereas for German, we used the monolingual data provided by the OPUS project. The quantitative statistics of these datasets is given in Table 2.

The blind test data provided by the organizers contained 1000 sentences each for translations between Lower Sorbian and German, 1000 sentences each for translations between Lower Sorbian and Upper Sorbian, and 1621 sentences each for translations between Upper Sorbian and German. We submitted the inferences on the blind test data to the shared task leaderboard.

3 Data Preparation

The data preprocessing step was crucial in determining the accuracies of our translations. The goal was not to waste resources (compute power, time) in processing things that don't add much value to extracting the semantics and understanding the text. (Tabassum and Patil, 2020)

For supervised learning, we preprocessed the source and target text using fairseq-preprocess[8], an inbuilt preprocessing script provided by Fairseq. We set the number of parallel workers for preprocessing the text as 20, so as to achieve faster preprocessing. Normalization (Mansfield et al., 2019) and pre-tokenization of the text is done before passing the to fairseq-preprocess. We used sacremoses[9]

[4] https://github.com/facebookresearch/XLM

[5] https://opus.nlpl.eu/

[6] https://github.com/glample/fastBPE

[7] https://www.witaj-sprachzentrum.de/

[8] https://fairseq.readthedocs.io/en/latest/command_line_tools.html

[9] https://github.com/alvations/sacremoses

tokenizer, which helps us to tokenize and normalize text according to our needs. fairseq-preprocess binarizes the training data and builds vocabularies from the text of that particular language.

For unsupervised learning, we applied some additional preprocessing, which consisted of using XLM-Moses tokenizer. The XLM-Moses tokenizer performs the following steps: removing unicode punctuations, normalizing punctuations (punctuations will be removed from the utterances during training) and removing any non-printing characters. We also applied byte-pair encoding (Sennrich et al., 2016) to our data, where we use the inbuilt script provided by fastBPE[10]. Byte-pair encoding algorithm computes the unique set of words used in the corpus (after the normalization and pre-tokenization steps are completed), and then builds the vocabulary by taking all the symbols used to write those words. Byte-pair encoding algorithm application includes learning the BPE codes from the training dataset, then applying the same on the training, validation and test datasets. Also, we get the training vocabulary once we have obtained the codes after training. Finally, we apply XLM preprocessing provided by Facebook XLM, to get the final preprocessed data.

4 Model Description

4.1 Supervised Training

We trained transformer models from scratch using the 'transformer' architecture provided by open-source toolkit Fairseq. Fairseq provides multiple state-of-the-art architectures to build translation models. Transformers use self-attention along with an encoder-decoder approach to train (Sutskever et al., 2014b). Encoders extract features from input sentences, and decoders use those features to produce output translations. The encoder in the transformer consists of multiple encoder blocks. Input sentences pass through encoder blocks, and the outputs of the last encoder block become the inputs to the transformer decoder. The decoder also consists of multiple decoder blocks, and feature information is received from the encoder by each block of the decoder.

4.2 Unsupervised Training

For unsupervised training, a masked language model (MLM) is implemented using data from both languages (source and target). We use the XLM model for easier implementation of the MLM objective. Masked prediction is implemented during the training steps, along with denoising autoencoding (Vincent et al., 2008), which involves reconstructing the original text data from a corresponding noisy version. We use an encoder-decoder transformer model, consisting of 12 layers in total (6 each to encoder and decoder), and is similar to the XLM architecture. We transfer the masked language model trained encoder transformer to the aforementioned encoder-decoder translation model.

5 Experiments

5.1 Training Details

For training the models we used the fairseq, a sequence model toolkit written in Pytorch (Paszke et al., 2019) developed by Facebook Artificial Intelligence Research (FAIR) team. We trained our models on the Nvidia Tesla K80 GPU, which has a 13GB RAM capacity.

For supervised learning, we trained our models on 50 epochs, and the total training time for every model was around 2 hours. We used the Adam optimizer (Kingma and Ba, 2014) for enhancing training performance, with corresponding beta coefficients set to 0.9 and 0.98. Label smoothing rate (Paszke et al., 2019) for the model is set to 0.1 (Label smoothing encourages the model to produce a finite output, which may lead to better generalization and prevent overfitting). Clip threshold of gradients is set to 0 (Zhang et al., 2019). A dropout (Srivastava et al., 2014) of 0.2 for input features is specified in the architecture. Maximum number of tokens in a batch is set to 4096 during training. Learning rate (Igiri et al., 2021) for the model is set to 0.0005.

For the unsupervised training, we train our models on 20 epochs, taking about five hours to train. Input words to the model are randomly shuffled during training, 3 at a time (Malkin et al., 2021). A word dropout of 0.1 is specified. 8 attention heads are taken in each layer of the encoder. Overall dropout and attention dropout of 0.1 is specified. 1000 tokens are taken per batch, and a batch-size of 32 is taken for training the models. Dimension of the embedding layer in the model was set to 1024. Sequence length of 256 is specified during training. We use the GeLU activation function (Hendrycks and Gimpel, 2016) in this model, instead of the typi-

[10] https://github.com/glample/fastBPE

Training Approach	**de-dsb**		**dsb-de**		**dsb-hsb**		**hsb-dsb**		**de-hsb**		**hsb-de**	
	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF
Supervised	20.8	44.1	25.4	51.3	49.1	65.5	50.7	66.9	25.7	49.1	29.7	53.8
Unsupervised	0.2	8.1	0.1	5.0	10.4	48.6	9.3	44.2	0.5	14.3	0.3	13.6

Table 3: Scores received on the translations obtained by performing supervised and unsupervised approaches for the WMT22-Unsupervised and Very Low Resource Supervised Task (de: German, dsb: Lower Sorbian, hsb: Upper Sorbian).

cal ReLU function used. Here too, Adam optimizer was used, with corresponding beta coefficients set to 0.9 and 0.98.

5.2 Evaluation Metrics

The BLEU (Papineni et al., 2002) and chrF (Popović, 2015) metrics were used for evaluation of the generated translations. The same metrics were used for evaluation on the shared task leaderboard.

BLEU stands for Bilingual Language Understudy. BLEU algorithm is used to evaluate machine translation quality. BLEU metric is language independent, and is easy to understand and compute. Higher the BLEU, better are the translations.

chrF stands for "character n-gram F-score". Informally, it measures the amount of overlap of short sequences of characters (n-grams) between the MT output and the reference. According to (Mathur et al., 2020) , chrF is "is technically the macro-average of n-gram statistics over the entire test set".

6 Results

For the results, please refer to Table 3. Table contains the BLEU and chrF scores to the translations that we obtained on all six language pairs (de-dsb, dsb-de, dsb-hsb, hsb-dsb, de-hsb, hsb-de), by both supervised and unsupervised approaches. These scores are obtained from our submissions to the leaderboard for the Unsup-Very Low Sup Shared task.

7 Related Work

Machine translation has been a pivotal field of research in the natural language processing domain. With rule-based and statistical machine translation methods proposed in the past decades, neural machine translation has surpassed these conventional methods by achieving state-of-the-art accuracies with each year. In 2014, Bahdanau (Bahdanau et al., 2015) proposed the base paper for neural machine translation. According to the paper, the encoder part of the model encodes the input sentence into a fixed length vector, from which the decoder generates the translation. The encoder and decoder parts could be neural architectures such as a simple RNN, LSTM (Sherstinsky, 2020), Bidirectional RNN (Schuster and Paliwal, 1997), GRU (Chung et al., 2014), etc. With the introduction of transformers (Vaswani et al., 2017) and self-attention in training neural networks, NMT research got a substantial boost.

German, being pretty high resource in nature; there has been significant work carried out in German in NMT. The Workshop on Machine Translation (WMT) has a significant contribution to the same. Minh-Thang Luong (Luong et al., 2015) demonstrate two seperate attention mechanisms (global and local attention) for bidirectional translations between English and German, gaining an increase of 5.0 in the BLEU score over non-attention based techniques. Macketanz (Macketanz et al., 2021) present the result of applying a fine-grained test suite on the outputs of 36 state-of-the-art machine translation systems between English and German, which were submitted to the Sixth Conference on Machine Translation. Xu (Xu et al., 2021) proposed BiBERT, a bilingual BERT model which helped in achieving state-of-the-art translation performance compared to other published papers till date, and that too without implementing backtranslation (Edunov et al., 2018). The paper also proposes a stoicastic layer selection method which helps in improving translation performance.

Sorbian family of languages have started receiving attention with regards to NMT research in the past few years. Li and team (Li et al., 2020) worked on supervised machine translation for a few language pairs, which included German-Upper Sorbian translations. They experimented with document-enhanced NMT, XLM pretrained language model enhanced NMT, etc. Their primary submissions won the first place in the German to Upper Sorbian Translation directions.

Knowles and team (Knowles et al., 2020) worked on implementing ensemble learning in transformer models for German-Upper Sorbian, built using BPE-dropout, lexical modifications and backtranslation.

Pertaining to unsupervised learning: Chronopoulou (Chronopoulou et al., 2020) propose the LMU Munich System for the WMT 2020 Unsupervised Machine Translation task, which involves using a pretrained monolingual model and finetuning it on both German and Upper Sorbian. Finally, the system uses backtranslation, and uses the pseudo-parallel data obtained to finetune the model further. Finally, the paper ensembles the best best-performing systems and give state-of-the-art scores on unsupervised translations between German and Upper Sorbian. Edman (Edman et al., 2021) implement transformer encoder-decoder architectures for unsupervised NMT from German to Lower Sorbian. The system has three modifications from the conventional methodology- training followes a bilingual approach, instead of a multilingual system approach. Secondly, a novel method is introduced for building the vocabulary of an unseen language. Finally, experimentation is done with the order of implementation of online and offline backtranslation. The paper received first place in the Unsupervised Machine Translation Task for WMT 2021.

8 Conclusion

Thus, we have implemented supervised and unsupervised neural machine translation approaches for translation between language pairs consisting of German(de), Lower Sorbian(dsb), and Upper Sorbian(hsb). We utilized different architectures for implementing the same. Our future plans include training these models with much larger corpora on computationally-efficient machines to obtain better evaluation metric scores and use high-end GPUs for practical training. We plan to use better preprocessing techniques and linguistic methods to improve the usefulness of the final training data to be fed to the models. We plan to implement backtranslation to improve current translation accuracy, have longer pre-training, and implement other pretrained models such as mBERT and XLM.

Acknowledgements

We are grateful to Dr. Geetanjali Kale and Dr. P. T. Kulkarni for their constant guidance. We also thank SCTR's Pune Center for Analytics with Intelligent Learning for Multimedia Data for their continuous support.

References

Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. 2017. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

G. Bebis and M. Georgiopoulos. 1994. Feed-forward neural networks. *IEEE Potentials*, 13(4):27–31.

Alexandra Chronopoulou, Dario Stojanovski, and Alexander Fraser. 2021. Improving the lexical ability of pretrained language models for unsupervised neural machine translation. In *NAACL-HLT*, pages 173–180.

Alexandra Chronopoulou, Dario Stojanovski, Viktor Hangya, and Alexander Fraser. 2020. The LMU Munich System for the WMT 2020 Unsupervised Machine Translation Shared Task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1084–1091, Online. Association for Computational Linguistics.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

M. Durrell. 2006. *Germanic Languages*, pages 53–55.

Lukas Edman, Ahmet Üstün, Antonio Toral, and Gertjan van Noord. 2021. Unsupervised translation of German–Lower Sorbian: Exploring training and novel transfer methods on a low-resource language. In *Proceedings of the Sixth Conference on Machine Translation*, pages 982–988, Online. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Chris Chinenye Emezue and Femi Pancrace Bonaventure Dossou. 2020. FFR v1.1: Fon-French neural machine translation. In *Proceedings of the The Fourth Widening Natural Language Processing Workshop*, pages 83–87, Seattle, USA. Association for Computational Linguistics.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus).

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80.

Phil Howson. 2017. Upper sorbian. *Journal of the International Phonetic Association*, 47(3):359–367.

Chinwe Igiri, Anyama Uzoma, and Abasiama Silas. 2021. Effect of learning rate on artificial neural network in machine learning. *International Journal of Engineering Research*, 4.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Rebecca Knowles, Samuel Larkin, Darlene Stewart, and Patrick Littell. 2020. NRC systems for low resource German-Upper Sorbian machine translation 2020: Transfer learning with lexical modifications. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1112–1122, Online. Association for Computational Linguistics.

Bin Li, Yixuan Weng, Fei Xia, and Hanjun Deng. 2022. Towards better chinese-centric neural machine translation for low-resource languages. *ArXiv*, abs/2204.04344.

Zuchao Li, Hai Zhao, Rui Wang, Kehai Chen, Masao Utiyama, and Eiichiro Sumita. 2020. SJTU-NICT's supervised and unsupervised neural machine translation systems for the WMT20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 218–229, Online. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Vivien Macketanz, Eleftherios Avramidis, Shushen Manakhimova, and Sebastian Möller. 2021. Linguistic evaluation for the 2021 state-of-the-art machine translation systems for German to English and English to German. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1059–1073, Online. Association for Computational Linguistics.

Nikolay Malkin, Sameera Lanka, Pranav Goel, and Nebojsa Jojic. 2021. Studying word order through iterative shuffling.

Courtney Mansfield, Ming Sun, Yuzong Liu, Ankur Gandhe, and Björn Hoffmeister. 2019. Neural text normalization with subword units. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 190–196, Minneapolis, Minnesota. Association for Computational Linguistics.

Nitika Mathur, Tim Baldwin, and Trevor Cohn. 2020. Tangled up in bleu: Reevaluating the evaluation of automatic machine translation evaluation metrics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Alex Sherstinsky. 2020. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404:132306.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014a. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014b. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Ayisha Tabassum and Dr. Rajendra R. Patil. 2020. A survey on text pre-processing & feature extraction techniques in natural language processing.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Theo Vennemann. 2008. Lombards and consonant shift: A unified account of the high germanic consonant shift. pages 213–256.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1096–1103, New York, NY, USA. Association for Computing Machinery.

Haoran Xu, Benjamin Durme, and Kenton Murray. 2021. Bert, mbert, or bibert? a study on contextualized embeddings for neural machine translation. pages 6663–6675.

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. 2019. Why gradient clipping accelerates training: A theoretical justification for adaptivity.

MUNI-NLP Systems for Lower Sorbian-German and Lower Sorbian-Upper Sorbian Machine Translation @ WMT22

Edoardo Signoroni and **Pavel Rychlý**
Faculty of Informatics
Masaryk University
e.signoroni@mail.muni.cz, pary@fi.muni.cz

Abstract

We describe our neural machine translation systems for the WMT22 shared task on unsupervised MT and very low resource supervised MT. We submit supervised NMT systems for Lower Sorbian-German and Lower Sorbian-Upper Sorbian translation in both directions. By using a novel tokenization algorithm, data augmentation techniques, such as Data Diversification (DD), and parameter optimization we improve on our baselines by 10.5-10.77 BLEU for Lower Sorbian-German and by 1.52-1.88 BLEU for Lower Sorbian-Upper Sorbian.

Introduction

This paper describes our Machine Translation (MT) systems for the WMT22 shared task on "Unsupervised MT and Very Low Resource Supervised MT"[1], which features translation between Lower Sorbian, Upper Sorbian, and German. Lower (*dsb* and Upper Sorbian (*hsb*) are Slavic minority languages spoken in the Eastern part of Germany with 7.000 and 30.000 native speakers respectively. Text data for these languages collected and made available by the Sorbian Institute and the Witaj Language Centre (Libovický and Fraser, 2021).

We submit systems for Lower Sorbian-German and Lower Sorbian-Upper Sorbian in both translation directions. We focused on the supervised approach, using only the parallel data made available by the task organizers for all the above languages.

We were able to improve on our baselines by: i. employing a new tokenization algorithm, High Frequency Tokenizer (HFT) (Signoroni and Rychlý, 2022); ii. augmenting the original parallel data with the Data Diversification (DD) technique by (Nguyen et al., 2020); iii. tuning the architecture and the parameters of the models, such as encoder/decoder depth, number of attention heads, dropout, batch size, etc.

We employed HFT since it aims to obtain more meaningful subword dictionaries, while DD was chosen because it does not involve additional data apart from the original parallel corpus. Both this techniques are relevant when working with a limited amount of data.

This paper is structured as follows: Section 1 summarizes the data used in training; Section 2 outlines our methodology, introducing our novel tokenizer and the models we used; Section 3 sums up our final systems, while Section 4 relates and discusses the results of our experiments; Section 5 contains some final remarks.

1 Data

We experiment with Lower Sorbian-German and Lower Sorbian-Upper Sorbian translation, using only the parallel data provided for each pair.

The parallel data for the dsb-de and the dsb-hsb pairs consist of ∼40k and ∼62k sentences respectively. We use only these data, as the approach we decided to follow does not need additional data. After applying this method, our final corpus size for training is ∼360k sentences for dsb-de, and ∼560k for dsb-hsb.

2 Methodology

In this section, we first present briefly our novel tokenizer, High Frequency Tokenizer, or HFT. Then, we describe the architecture of our models and how we trained them.

2.1 High Frequency Tokenization

Sennrich and Zhang (2019) showed that a meaningful subword tokens vocabulary is crucial for achieving good performance in low-resource NMT. While they experiment with BPE, we employ our novel tokenization algorithm, High Frequency Tokenizer, or HFT. This word segmentation methods aims to provide more meaningful subword dictionaries by obtaining more frequent, and thus better

[1] https://statmt.org/wmt22/unsup_and_very_low_res.html

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1111 - 1116
December 7-8, 2022 ©2022 Association for Computational Linguistics

¦	<token-delimiter>
↑	<single-uppercase>
▁	<explicit-whitespace>
∇	<all-uppercase>
Δ	<end-of-uppercase>

Figure 1: Special characters in the pretokenization and tokenization.

represented subwords. Given the importance of tokenization for low-resource NMT, we argue that is an important point to consider.

HFT uses the advantage of pretokenization, where sentences are split into tokens on the borders of alphanumeric and non-alphanumeric characters. The current prototype uses the regular expression `\b` of the Unix `sed`[2] command . Both the beginning and the end of each token is explicitly annotated, differently from previous systems such as `subword-nmt` BPE and `sentencepiece`.

HFT subwords are learnt from these tokens, they never cross the token boundaries, each token from the pretokenization is handled independently from other tokens. It speeds up both vocabulary learning and actual subword tokenization.

We also use case normalization for characters with both uppercase and lowercase. A single uppercase letter is changed to a special <uppercase-next> character and lowercase version of the given letter. A sequence of uppercase letters is changed to lowercase with a special <all-uppercase> and <end-of-uppercase> characters attached to the beginning and the end of the sequence. Figure 1 gives the special characters `hft` uses in pretokenization and tokenization.

The learning algorithm starts from a vocabulary containing all characters from the training text as possible subwords and the number of occurrences of the given subword (character). Then, it gradually increase the vocabulary in the following steps:

1. it processes all the words (tokens) from the pretokenized text to find the best subword segmentation using only subwords from the current vocabulary, counts the frequencies of each subword and of all possible subword candidates (pairs of succeeding subwords);

2. selects the top K candidates with the highest frequency and adds them as new subwords to the vocabulary (K is 5% of the target vocabulary size as default);

3. removes from the vocabulary all non-single-character subwords with frequency lower than the last added candidate;

4. repeat from 1. until the requested vocabulary size is reached

The best subword tokenization (in step 1) searches in all possible subword segmentation sequences the one with the lowest number of tokens and, for same number of tokens, the highest minimum frequency.

We evaluated HFT against Byte-Pair Encoding (BPE) (Sennrich et al., 2016) and Unigram (Kudo, 2018) on the metrics described by (Gowda and May, 2020) and on a weighted average of the frequencies of the tokens in the vocabulary. HFT performed well, providing better results for almost all test cases. Preliminary data on HFT's impact on downstream NMT also showed promising results. (Signoroni and Rychlý, 2022)

Moreover, during the experimentation for this task, we further confirmed that using HFT-tokenized data leads to better translation quality, calculated with sacreBLEU (Post, 2018), against a `subword-nmt` BPE baseline.

2.2 Data Diversification

For our final models we follow the Data Diversification (DD) approach of Nguyen et al. (2020). While most of the research in low-resource NMT avails itself of external data by employing techniques such as backtranslation and transfer learning, this simple, yet effective method does not need any external data, but only the original parallel corpus. The DD procedure is the following:

1. Train k different models on the authentic parallel corpus, in both the forwards and backwards directions;

2. Infer the translations with all the trained models, so to obtain k synthetic source and target data;

3. merge the translations to create a new parallel dataset, which comprises the original parallel data, plus an authentic source to synthetic target, and a synthetic source to authentic target section;

[2] https://www.gnu.org/software/sed/manual/sed.html

```
↑¦mjazy¦ ¦nimi¦ ¦namak ajomy¦ ¦f ary¦ ,▁ ¦kjar c my¦ ,▁ ¦mły ny¦ ,▁ ¦kow alnje¦ ¦a¦ ¦šule¦ .
↑¦mjaz¦ ¦nimi¦ ¦jo¦ ¦tek¦ ¦górnoserb ski¦ △¦sa ek¦▽ ¦wid e o kr už k¦ ¦gromaźe¦ ¦z¦ ▁„ ↑¦ign
```

Figure 2: Sample of Lower Sorbian text tokenized with HFT.

4. Train new models on these augmented parallel data;
5. repeat for *n* rounds.

Since Nguyen et al. (2020) reports that additional rounds of DD do not boost the performance of the resulting systems significantly, our systems were trained after just one round of DD. We have a level of diversification *k*=4, since we included all the previous experiments models' output, plus the original parallel data.

2.3 Preprocessing

We use both training and development test data as provided, without cleaning of any kind. We tokenize the data with two subword tokenization algorithms, BPE and HFT. For the former, we use the `subword-nmt` implementation. For HFT, we use our own implementation. We experiment with different vocabulary sizes, and our results are in line with previous research, such as Sennrich and Zhang (2019). They showed that a smaller vocabulary sizes improves the performance of low-resource NMT. In line with these findings, our experiments with vocabulary size of 12k and 10k, even if well below the standard 32k, performed worse than our final choice of 4k tokens.

We train a tokenizer for each language on the train split of the datasets, and share the dictionaries during all the stages of training.

2.4 Models

Table 1 gives details about the architecture and training parameters of each model we trained.

We experiment with two different model architectures, both based on the Transformer (Vaswani et al., 2017). The first, which we dubbed t-[tok][3], is a standard Transformer (Vaswani et al., 2017); while the second, called and t-opt-[tok], is a Transformer with optimized parameters for the size of the dataset.

We use Fairseq (Ott et al., 2019) for training the models, generating translations, and evaluating them.

As our baseline, we train a Transformer (Vaswani et al., 2017) with default hyper parameters and BPE tokenization, which we refer to as t-bpe. [4] As a first experiment, we train t-hft, a standard Transformer trained on data tokenized with HFT. We use *adam* as optimizer and we maximize BLEU score on the validation set at each epoch. For the BPE models, we use detokenized BLEU, but for HFT this was not implemented during training. We train both t-[tok] models for 100 epochs with dropout of 0.1, and 10240 maximum tokens for each batch. We use a learning rate of 0.0005 and the inverted square root scheduler for all of our models.

Secondly, we train another Transformer with the optimized hyper parameters found by (Araabi and Monz, 2020) for a dataset of 40k sentence pairs: 5 encoder/decoder layers, 2 attention heads, and a feed-forward dimension of 2048. During our experiments, however, we observed that a feed-forward dimension of 1024 gives better results. We do this with data tokenized with both methods, obtaining t-opt-bpe and t-opt-hft. These models are trained for 100 epochs, with dropout of 0.3, label smoothing of 0.5, encoder and decoder word dropout of 0.1, activation dropout of 0.3, and a maximum batch size of 4096 tokens.

Lastly, we build the DD parallel corpus by collating the outputs of all previous systems in both directions, beginning with the authentic parallel data, and adding both combinations of original and synthetic source and target data. We then use the DD-data to train both t-bpe-dd and t-hft-dd, which share the same architecture the t-[tok] models. The training is also similar, just differing in the number of epochs, which for these last models is 50.

2.5 Evaluation

To find our best candidates for submission, we generate translation on a development test set of unseen sentence pairs, either provided by the task organizers, or set aside from the train portion of the data. We produce translations with the standard settings (beam search with a beam of 5) using the best

[3] We trained models on data tokenized both with BPE and HFT. Since they share the same architecture and training parameters, [tok] stands either for bpe or hft.

[4] We still use a vocabulary size of 4k, which is already an improvement on the standard size of 32k.

PARAMETER	MODEL		
	t-[tok]	t-opt-[tok]	t-[tok]-dd
Vocabulary size	4000	4000	4000
feed-forward dimension	2048	1024	2048
attention heads	8	2	8
dropout	0.1	0.3	0.3
enc/dec layers	6	5	6
label smoothing	0.1	0.5	0.1
enc/dec word dropout	0.0/0.0	0.1/0.1	0.0/0.0
activation dropout	0.0	0.3	0.0
max tokens	10240	4096	10240

Table 1: Architecture details and training parameters for each model.

	DSB-DE	DE-DSB	DSB-HSB	HSB-DSB
t-bpe	27.92	22.74	72.01	69.71
t-hft	34.20	30.86	72.21	70.71
t-opt-bpe	29.75	25.06	71.37	69.50
t-opt-hft	35.46	31.12	71.83	68.95
t-bpe-dd	33.02	28.54	73.47	71.98
t-hft-dd	38.42	33.53	73.53	71.59

Table 2: Trained models and BLEU score during inference on development test data

checkpoint of each model and evaluate the detokenized output with sacreBLEU (Post, 2018). For the BPE models, we detokenize with the provided argument, while for HFT we use our own plug-in script. We use the same settings to translate the test set for our submissions. Table 2 gives BLEU scores, computed on the development test sets, for each system we experimented with.

During inference on the development test set, models trained on HFT data outperformed the BPE baseline by 4.99 to 8.12 BLEU for the dsb-de pair, while for dsb-hsb the difference in score is minimal. t-opt-[tok] was better than the corresponding t-[tok] model in the dsb-de pair. For dsb-hsb, this does not hold true, with t-opt-[tok] always performing worse than the baseline. t-[tok]-dd improves on both the baseline and t-opt-[tok] for every language pair and direction.

2.6 Inconclusive and Negative Results

During DD, we collated data from all four experimental models for each pair, both t-[tok] and t-opt-[tok], regardless of their performance. This later resulted in our best systems. For the dsb-de pair, we also tried to ensemble data from the four best performing systems to create the DD train set, all from HFT data and ranging from 34.99 to 37.20 BLEU on the dsb-de side, and 31.12 to 30.42 on the reverse direction. This was done with the intuition that better train data should result in better performance. However, after training t-[tok]-dd on these data, the resulting system that performed worse by -0.58, giving 37.84 BLEU on the development test set. In contrast, the final t-[tok]-dd gave us 38.42 BLEU, and was trained on data generated with systems ranging from 27.92 to 35.46 BLEU on the dsb-de side, and from 22.74 to 31.12 for the reverse direction.

While this small difference in BLEU score may not be significative, further investigation should be conducted as this may indicate that a more diverse dataset is better than one with a higher quality for training with NMT systems with DD. Our initial hypothesis for why this happens is that being the systems' performance closer, the translations they generate are similar. This leads to worse generalization potential for the resulting final system.

3 Final Systems

Table 3 gives BLEU and chrF scores for our final submissions.

For all pairs and directions we worked on, our best systems was t-hft-dd, a Transformer trained on a single NVIDIA A40[5] for 50 epochs on DD

[5] Previous systems were always trained on a single GPU,

		t-hft-dd
DSB-DE	BLEU	49.5
	chrF	73.0
DE-DSB	BLEU	50.5
	chrF	74.1
DSB-HSB	BLEU	72.2
	chrF	87.5
HSB-DSB	BLEU	72.3
	chrF	87.5

Table 3: BLEU and chrF scores for our best systems on the final test set.

HFT-tokenized data, with a vocabulary size of 4k, feed-forward dimension of 2048, dropout 0.3, and 10240 for each batch.[6]

Our hypotheses on why HFT leads to improvements on these datasets are the following. On top of providing more frequent and better defined token for the model to learn, it also explicitly marks both beginning and end of the words during pretokenization. This could be relevant for morphologically complex languages, such as the ones in this task, since it provides more information to the model on possible prefixes and suffixes. Contrast this with the fact that `subword-nmt` BPE only explicitly mark continuation with the @@ marker. This kind of tokenization thus makes no distinction between full words and word endings. Moreover, it seems to struggle with capitalized words and punctuation, which can also be informative, if handled optimally. HFT's pretokenization seems to help with this issue. Investigating these topics will be further addressed by future work.

4 Conclusions

This paper described our submission for the WMT22 shared task on Unsupervised MT and Very Low Resource Supervised MT. We presented systems for Lower Sorbian-German and Lower Sorbian-Upper Sorbian translation in both direction under supervised training conditions. To train our best systems we employed a novel tokenization algorithm, HFT, to obtain more meaningful subword vocabularies, contrasted to a BPE baseline; and Data Diversification (Nguyen et al., 2020) to augment the training data using only the parallel dataset provided for each language pair.

During our experiments we confirmed that optimizing not only the Transformer's hyper parameters, but also the subword vocabulary quality and size, are crucial steps for low-resource NMT. Choosing the appropriate vocabulary size for the dataset, could lead to significant improvements in BLEU score even with a small amount of parallel data. These, however, are still open and complex problems, since previously proposed settings or, even more so default ones, did not always provide the best results.

Limitations and Future Work

While providing NMT systems for Lower Sorbian-German and Lower Sorbian-Upper Sorbian that perform reasonably well, some other methods, some of which were used by other submissions, could provide better results. These should be taken into account, if a system for these language pairs will be deployed for language conservation, revitalization, or everyday use. Such system may also be hampered by the limited scope of the training data, which is inherent in their size. As for other low-resource and, especially for endangered languages, documentation ventures such as those of the Sorbian Institute and the Witaj Language Centre are vital to create bigger, more comprehensive datasets, which are still needed for the current NLP methodologies to work at their best.

Ethics Statement

NMT systems are, as every other data-driven technology, sensible to biases and other shortcomings in their training data. De-biasing datasets and NLP systems' output is a scope of research that lies outside the scope of this shared task and thus, from the scope of this paper. If these systems were to be employed in a real-world scenario such as language conservation, revitalization or everyday translation, we advise caution as to the limitations mentioned above.

Following Lacoste et al. (2019), we report that the experiments and the research that led to the results presented in this paper were conducted on a private server infrastructure consisting of a NVIDIA Tesla T4, A40, and A100 for around 500 hours of training at an efficiency of 0.59 kg/kWh[7] for a total of 20.39 kg CO_2 eq.

variably on a A40, A100 or a Tesla T4. Training times did not exceed 12 hours for both t-[tok]-dd systems.

[6] Every other unmentioned parameter was left at the default setting.

[7] The Czech Republic's country average as reported in https://www.carbonfootprint.com/docs/2018_8_electricity_factors_august_2018_-_online_sources.pdf

Acknowledgements

We thank the reviewers for their useful comments. The work of the authors is partly supported by the Internal Grant Agency of Masaryk University, by Lexical Computing and by the Ministry of Education of the Czech Republic within the LINDAT-CLARIAH-CZ project LM2018101.

References

Ali Araabi and Christof Monz. 2020. Optimizing transformer for low-resource neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3429–3435, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Thamme Gowda and Jonathan May. 2020. Finding the optimal vocabulary size for neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3955–3964, Online. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*.

Jindřich Libovický and Alexander Fraser. 2021. Findings of the WMT 2021 shared tasks in unsupervised MT and very low resource supervised MT. In *Proceedings of the Sixth Conference on Machine Translation*, pages 726–732, Online. Association for Computational Linguistics.

Xuan-Phi Nguyen, Shafiq Joty, Wu Kui, and Ai Ti Aw. 2020. Data diversification: A simple strategy for neural machine translation. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

Edoardo Signoroni and Pavel Rychlý. 2022. HFT: High frequency tokens for low-resource NMT. In *Proceedings of the Fifth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2022)*, pages 56–63, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

The AIC System for the WMT 2022 Unsupervised MT and Very Low Resource Supervised MT Task

Ahmad Shapiro, Mahmoud Tarek, Omar Khaled
Mohamed Fayed, Ayman Khalafallah, Noha Adly
Applied Innovation Center
{ahmad.shapiro, mahmoudtarek, omar.khaled, m.essam, a.khalafallah, nadly}@aic.gov.eg

Abstract

This paper presents our submissions to WMT 22 shared task in the Unsupervised and Very Low Resource Supervised Machine Translation tasks. The task revolves around translating between German ↔ Upper Sorbian (de ↔ hsb), German ↔ Lower Sorbian (de ↔ dsb) and Upper Sorbian ↔ Lower Sorbian (hsb ↔ dsb) in a both unsupervised and supervised manner. For the unsupervised system, we trained an unsupervised phrase-based statistical machine translation (UPBSMT) system on each pair independently. We pretrained a German-Slavic mBART model on the following languages Polish (pl), Czech (cs), German (de), Upper Sorbian (hsb), and Lower Sorbian (dsb). We then fine-tuned our mBART on the synthetic parallel data generated by the (UPBSMT) model along with authentic parallel data (de ↔ pl, de ↔ cs). We further fine-tuned our unsupervised system on authentic parallel data (hsb ↔ dsb, de ↔ dsb, de ↔ hsb) to submit our supervised low-resource system.

1 Introduction

Like most machine learning approaches, data can be considered the most important component of the recipe for modeling a solution for a given problem. Neural machine translation relies heavily on a large amount of training data to correctly model two languages and learn the mapping between them to produce semantically and syntactically right translations. However, machine translation is not available for the majority of the 7000 languages spoken on the earth. This is due to the fact that parallel corpora are scarce or non-existent. There have been several proposals to alleviate the issue of small amounts of parallel data such as pivot translation, multilingual training, and semi-supervised training which resulted in an acceptable performance.

Unsupervised machine translation became the go-to solution when lacking parallel data. The WMT 2022 Unsupervised MT Task focuses on two very low-resource languages: Upper Sorbian (HSB) and Lower Sorbian (DSB). Upper and Lower Sorbian are minority languages spoken in the federal states of Saxony and Brandenburg in Eastern Germany. With just 30,000 and 7,000 native speakers, working on these languages is an extreme low-resource task, with little prospect of ever approaching the number of resources available for languages with millions of speakers. However, because they are western Slavic languages, the Sorbian languages can benefit from Czech and Polish data (Libovický and Fraser, 2021).

In this paper, we describe our systems for translating between German ↔ Upper Sorbian (de ↔ hsb), German ↔ Lower Sorbian (de ↔ dsb), and Upper Sorbian ↔ Lower Sorbian (hsb ↔ dsb) in a both unsupervised and supervised manner.

We approach the task by combining two novel approaches for unsupervised machine translation. Influenced by (Artetxe et al., 2019), we start by developing unsupervised phrase-based statistical machine translation systems (UPBSMT) for all language pairs independently. In contrast to (Artetxe et al., 2019), (Lample and Conneau, 2019) relies on pre-training an XLM model on the source and target language to capture the translation signal instead of using (UPBSMT). Instead, we benefit from both the pre-training and UPBSMT. So, we pre-train an mBART model (Liu et al., 2020) on Polish, Czech, Upper Sorbian, Lower Sorbian, and German from scratch as we mentioned earlier that pl and cs are similar to dsb and hsb. We then fine-tune mBART on synthetic parallel data (de),(de ↔ hsb) and (hsb ↔ dsb) along with authentic parallel data (de ↔ pl, de ↔ cs).

We group pl, cs, dsb, hsb under one token $slavic$ while feeding it to the encoder. For our low-resource submission, we fine-tune the unsupervised model on the authentic parallel data provided by the task between (de ↔ hsb),(de ↔ dsb), and (hsb ↔ dsb).

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1117 - 1121
December 7-8, 2022 ©2022 Association for Computational Linguistics

Our unsupervised approach scored the highest BLEU in all directions except (de ↔ dsb) direction.

2 Related Work

The earliest approach on Unsupervised Machine Translation was introduced by (Ravi and Knight, 2011) where they frame the MT task as a decipherment task, treating the target language as cipher text of English. Their method is essentially the same approach taken by cryptanalysts and epigraphers when they use the source texts. They started by estimating the word translation probabilities using a devised Iterative EM algorithm, due to the huge consumption of memory since they operate on large-scale vocabularies. Followed by that, they propose a novel approach based on Bayesian Decipherment that outperformed the previous EM approach in all aspects. After that they build an n-gram translation table that was used to estimate an IBM Model 3 translation model, the highest BLEU (Papineni et al., 2002) score achieved was 19.3 on the Spanish-English OPUS subtitles data.

(Artetxe et al., 2019) provide a two-step solution to unsupervised machine translation. For step one, they start by building an UPBSMT system between source and target languages. Resulting in two translation models: source-to-target and target-to-source models. Using these models, they back-translate target monolingual data using the target-to-source model to generate $(\hat{src}, trg)$ pairs that will be used to train the source-to-target neural model. Similarly, they back-translate the source monolingual using the source-to-target model to generate $(src, \hat{trg})$ pairs that will be used to train the target-to-source neural model.

The second step is training two neural models, source-to-target and target-to-source, using the synthetic data generated from step 1 using iterative back-translation. The first iteration relies solely on data generated by UPBSMT, the following iterations substitute a percentage of synthetic data generated by UPBSMT by back-translated data from the neural model in the reverse direction. Until the whole training data is back-translated from the reverse model. In contrast, (Lample and Conneau, 2019) starts by pre-training an XLM encoder on Masked Language Modeling (MLM) task on the source and target languages.

After pre-training, they initialize an encoder-decoder model using the pre-trained XLM encoder. Their training step is composed of three tasks :

1. Denoising Auto encoding.
2. Cross Domain (Back-translation).
3. Adversarial Loss.

In our work, we combine the two methods. But instead of using neural iterative back-translation, we add authentic parallel data from related languages.

3 Approach

Inspired by (Artetxe et al., 2019) and (Lample and Conneau, 2019), we adapted a mixed approach to mitigate the weaknesses and combine the advantages of both methods. (Artetxe et al., 2019) use UPBSMT as an explicit initial translation signal to train two translation models from scratch on a translation task. But, UPBSMT's output is noisy and the translation model is trained from scratch without any denoising pre-training objective. In contrast, (Lample and Conneau, 2019) pre-trains an XLM encoder on MLM task and then use it to initialize a seq2seq model which will be trained to translate in an unsupervised manner as we discussed in Section 2. Although (Lample and Conneau, 2019) didn't train the translation model from scratch, they relied solely on the three training tasks discussed earlier to capture the cross-lingual translation signal in contrast to (Artetxe et al., 2019) who used an explicit cross-lingual translation signal.

We combine the best of both worlds by using UPBSMT as our initial translation signal to fine-tune a pre-trained mBART model on a multilingual translation task.

3.1 Unsupervised Phrase-based Statistical Machine Translation

We followed (Artetxe et al., 2018) approach to build an unsupervised phrase-based statistical machine translation system between the following pairs : $(de \rightarrow dsb)$, $(de \rightarrow hsb)$, $(dsb \rightarrow de)$, $(hsb \rightarrow de)$, $(hsb \rightarrow dsb)$ and $(dsb \rightarrow hsb)$.

Using the above models, we back-translated monolingual data of $lang_1$ to $\hat{lang_2}$ which will be used to train the reverse direction model as following :

1. de translated by $(de \rightarrow dsb)$ model, producing $(\hat{dsb}, de)$ pairs to train the $(dsb \rightarrow de)$ neural direction.
2. de translated by $(de \rightarrow hsb)$ model, producing $(\hat{hsb}, de)$ pairs to train the $(hsb \rightarrow de)$ neural direction.

3. dsb translated by $(dsb \rightarrow de)$ model, producing $(\hat{de}, dsb)$ pairs to train the $(de \rightarrow dsb)$ neural direction.

4. hsb translated by $(hsb \rightarrow de)$ model, producing $(\hat{de}, hsb)$ pairs to train the $(de \rightarrow hsb)$ neural direction.

5. dsb translated by $(dsb \rightarrow hsb)$ model, producing $(\hat{hsb}, dsb)$ pairs to train the $(hsb \rightarrow dsb)$ neural direction.

6. hsb translated by $(hsb \rightarrow dsb)$ model, producing $(\hat{dsb}, hsb)$ pairs to train the $(dsb \rightarrow hsb)$ neural direction.

3.2 German-Slavic mBART pre-training

Since Lower and Upper Sorbian are West-Slavic languages, their direct cousins in the West-Slavic family tree are Polish (pl) and Czech (cs). Polish and Czech are high-resource languages with a large-scale availability of both monolingual and parallel data. We pre-trained mBART model (Liu et al., 2020) from scratch on denoising auto-encoding objective on Polish (pl), Czech (cs), Upper Sorbian (hsb), Lower Sorbian (dsb), German (de).

3.3 mBART fine-tuning

Using the generated synthetic parallel data produced from UPBSMT step discussed in Section 3.1 along with authentic $(pl \rightarrow de)$ and $(cs \rightarrow de)$ from OPUS (Tiedemann, 2012). We fine-tuned our German-Slavic mBART on translation on $(pl \rightarrow de)$, $(cs \rightarrow de)$, $(de \leftrightarrow dsb)$, $(de \leftrightarrow hsb)$, $(hsb \leftrightarrow dsb)$. Taking advantage of the similarity between (pl, cs, dsb, hsb), we grouped those languages under one language token $(slavic)$ which is fed to the encoder of our mBART. This approach constructs our unsupervised submission.
For the low-resource submission, we further fine-tuned the resulted model on authentic parallel data provided by the task.

4 Experiments

In this section, we describe our experimental setup and results. Readers can refer to our GitHub Repository [1] for training scripts, checkpoints, hyper-paramters etc.

4.1 Data Pre-processing

We follow (Artetxe et al., 2019) cleaning approach as following :

1. `normalize-punctuation.perl` script from Moses library to normalize punctuations.

2. `remove-non-printing-char.perl` script from Moses library to remove non-printing characters.

3. Tokenizing using Moses Tokenizer.

4. Deduplication.

5. Cleaning by length, with minimum and maximum of 3 and 80 words respectively.

Language	Datasets	Sentences
Polish (pl)	europarl-v10	706,047
	news-crawl 2018 to 2021	12,653,333
	Total	13,359,380
Czech (cs)	europarl v10	669,676
	news-commentry v14-16	825,841
	news-crawl 2007 to 2021	109,599,883
	Total	111,095,400
German (de)	europarl v10	2,107,971
	news-commentry v14-16	1,259,790
	news-crawl 2007 to 2021	428,057,920
	Total	431,425,681
Upper Sorbian (hsb)	Witaj (2020)	222,027
	Sorbian-Insitute (2020)	339,822
	Task Data (2022)	436,579
	Total	998,428
Lower Sorbian (dsb)	Task Data (2021)	145,198
	Task Data (2022)	66,407
	Task Data : Wiki (2022)	8,814
	Total	220,419

Table 1: Monolingual Data sets used in our experiments

4.2 Unsupervised Statistical Machine Translation Data

We use monolingual data of German, Upper Sorbian and Lower Sorbian stated in Table 1. We used a 20MILL random sample from the German monolingual data. The output of the UPBSMT is synthetic parallel data that will be used to fine-tune the pre-trained mBART on the unsupervised translation task. The number of synthetic parallel data is shown in Table 2.

[1] https://github.com/ahmadshapiro/WMT22

Language	Sentences
dsb → de	19,486,715
de → dsb	155,683
hsb → de	19,486,715
de → hsb	873,794
hsb → dsb	873,794
dsb → hsb	155,683

Table 2: Synthetic Parallel Data Generated by UPBSMT

4.3 mBART Pre-training

We pretrained mBART on 32 V100 GPUs from scratch for less than 2 epochs (24hrs) on all monolingual data from Table 1. We learned 32k BPE codes using SentencePiece Library (Kudo and Richardson, 2018) on the concatenation of all monolingual data. This SentencePiece model will be used for the rest of neural experiments involving mBART. The average valid perplexity for all languages reached 4.16. We decided to stop training due to the time limit. All of our neural models were developed using FairSeq Framework (Ott et al., 2019).

4.4 mBART Fine-tuning (Unsupervised Submission)

We fine-tuned our pre-trained mBART on translation task using authentic parallel data of (pl-de, cs-de) shown in Table 3 along with all synthetic parallel data shown in Table 2. We grouped (hsb, dsb, pl, cs) under one token (slavic) which is passed to mBART encoder as we discussed earlier in Section 3.3. The training was done on 27 V100 GPUs for less than 1 epoch (24hrs).

4.5 mBART Fine-tuning (Low Resource Submission)

We further fine-tuned mBART on authentic parallel task data of (hsb-de, dsb-de, hsb-dsb) shown in Table 3 for 3 epochs to submit our supervised model.

5 Results

In this section, we present our results on the blind test set of WMT 22 workshop.

5.1 Unsupervised Submission

Our approach scored the highest BLEU in all pairs except the (de ↔ dsb) directions. This can be attributed to the fact of having multiple errors in the UPBSMT experiment on this specific pair. Due to the time limit, we had to use the un-tuned/corrupted models for this pair. In contrast, (de ↔ hsb) directions models scored almost 18.0 BLEU score. Surprisingly, this can reflect the importance of the UPBSMT component in our experiments, since hsb and dsb are hugely similar. But, due to an error in the UPBSMT training, the former hugely outperformed the latter. Results are reported in Table 4.

Language	Datasets	Sentences
pl-de	DGT JRC-Acquis MultiParaCrawl EUbookshop Europarl QED	12,375,574
cs-de	DGT JRC-Acquis MultiParaCrawl EUbookshop Europarl QED	12,427,403
hsb-de	Task Data (2020)	60,000
	Task Data (2021)	87,521
	Task Data (2022)	301,536
	Total	448,787
dsb-de	Task Data (2022)	40,193
hsb-dsb	Task Data (2022)	62,564

Table 3: Authentic Parallel Data sets from OPUS (Tiedemann, 2012) used in our experiments

Direction	BLEU
dsb → de	4.0
de → dsb	1.2
hsb → de	**18.0**
de → hsb	**17.9**
hsb → dsb	**35.9**
dsb → hsb	**44.2**

Table 4: Unsupervised results on Blind Test data of WMT22

5.2 Low Resource Submission

As shown in Table 5, further fine-tuning on authentic parallel data improved BLEU score in all directions even the corrupted (de ↔ dsb) directions. Our model was constantly improving through updates, but we had to stop the training due to time constraints. We didn't use any low-resource techniques such as back-translation, BPE dropout, etc.

Direction	**BLEU**
dsb → de	39.4
de → dsb	48.2
hsb → de	47.5
de → hsb	51
hsb → dsb	66.6
dsb → hsb	65.8

Table 5: Supervised results on Blind Test data of WMT22

6 Conclusion and Future Work

In this paper, we describe our submission to the WMT 2022 shared task of Unsupervised and Very Low Resource Supervised Machine Translation. We combined the advantages and mitigated the weaknesses of two novel unsupervised approaches along with pre-training a German-Slavic mBART model. Ablation studies for different components of our approach are left for future work.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2019. An effective approach to unsupervised machine translation. *CoRR*, abs/1902.01313.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

Jindřich Libovický and Alexander Fraser. 2021. Findings of the wmt 2021 shared tasks in unsupervised mt and very low resource supervised mt. In *Proceedings of the Sixth Conference on Machine Translation*, pages 726–732, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Sujith Ravi and Kevin Knight. 2011. Deciphering foreign language. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 12–21, Portland, Oregon, USA. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218. Citeseer.

NICT at MixMT 2022: Synthetic Code-Mixed Pre-training and Multi-way Fine-tuning for Hinglish–English Translation

Raj Dabre
National Institute of Information and Communications Technology (NICT)
Kyoto, Japan
raj.dabre@nict.go.jp

Abstract

In this paper, we describe our submission to the Code-mixed Machine Translation (MixMT) shared task. In MixMT, the objective is to translate Hinglish to English and vice versa. For our submissions, we focused on code-mixed pre-training and multi-way fine-tuning. Our submissions achieved rank 4 in terms of automatic evaluation score. For Hinglish to English translation, our submission achieved rank 4 as well.

1 Introduction

Code-mixed translation is the task of translation involving code-mixed languages. A code-mixed language is one which combines words as well as grammar of two or more languages. Code-mixed translation is difficult because of the lack of training data for the same despite its ubiquitous usage. One widely used code-mixed language is Hinglish which combines Hindi and English. Hinglish sentences are typically constructed either by replacing some Hindi words or phrases with English ones in a Hindi sentence or vice versa. Sometimes, a sentence starts off in one language but ends in another. There are also complex cases where the grammatical structures of both languages are melded into one. Hinglish is typically written in Roman letters, although there are cases when it is written in Devanagari.

In this paper we describe our submissions to the MixMT task which involves Hinglish to English and English/Hindi to Hinglish translation. The main challenge of this task is that the parallel corpus available for training models is rather scarce. The total amount of clean, non-synthetic data available for MixMT is around 18,000 examples for both directions. Therefore, we have no choice but to rely on external sources of data, and use them to pre-train models. In our case, we leverage a large amount of Hindi–English parallel data and synthesize pseudo Hinglish data. To do this, perform word alignment on the Hindi–English data and then replace random English phrases with aligned Hindi phrases. We then use the synthetic Hinglish–English parallel data for pre-training. The pre-trained model is then fine-tuned to train a joint bidirectional Hinglish–English translation model. According to the automatic evaluation metrics, we obtain 4th rank and on human evaluation of Hinglish to English translation, we also obtain 4th rank. Unfortunately, for translation into Hinglish our system ends up copying the English inputs as outputs. Although automatic evaluation scores for this are reasonably high, their human evaluation scores are lowest since the sentences are not Hinglish at all.

2 Related Work

Work on code-mixed machine translation is relatively new, especially for Hinglish. Two important works in this regard are HinGE (Srivastava and Singh, 2021) which proposes a dataset for English/Hindi to Hinglish translation and PHINC (Srivastava and Singh, 2020) which proposes a dataset for Hinglish to English translation. The HinGE dataset contains natural as well as human rated synthetic examples in both Hindi and English as source languages. Having two sources is expected to help in Hinglish generation, as the model will have the advantage of contexts from both sources. In our case, we did not leverage both sources and focused only on English. On the other hand, PHINC is designed for Hinglish to English translation and is much larger than HinGE. Neither of these datasets are perfect and contain some noisy examples, but the lack of other datasets leaves us with no choice.

Due to lack of code-mixed data, it is natural to consider synthetic code-mixed data creation where Gupta et al. (2020) show that leveraging an XLM model (CONNEAU and Lample, 2019) and linguistic features can help generate high quality code-mixed sentences. However, we opted for a quicker way using word alignment and phrase substitution

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1122 - 1125
December 7-8, 2022 ©2022 Association for Computational Linguistics

approach. Using pre-trained models, can be very helpful in code-mixed translation as they are able to represent them effectively (Santy et al., 2021). Agarwal et al. (2021) have shown that pre-trained models (Liu et al., 2020) are highly effective, but we focused more on using our own models trained on our synthetic data.

Apart from machine translation, code-mix Hinglish has been reasonably explored for natural language understanding tasks, particularly for sentiment analysis. We refer interested readers to the following works: Baroi et al. (2020); Singh and Lefever (2020); Mathur et al. (2018); Bhange and Kasliwal (2020).

3 Methods

We describe the synthetic code mixed pre-training and multi-way fine-tuning approaches we used for our submissions.

3.1 Synthesizing Code-Mixed Data

We assume the existence of a large amount of Hindi–English parallel corpus, which we use to synthesize Hinglish. Since Hinglish is written in the Roman alphabet, we first Romanize it. We then use an aligner to obtain word alignments between Hindi and English. For each English sentence, we take a random span of tokens, find the corresponding aligned span of tokens in Hindi and replace it with the English tokens span. We note that this assumes that the language structure of Hindi is preserved in this process. To determine the span in the target language, we find the indices of the aligned target words and then choose the smallest as the starting index and the largest as the ending index as the span to be replaced. This is known as the min-max approach, which was used by Zenkel et al. (2021). As a result of this process we obtain a Hinglish–English parallel corpus where Hinglish is synthetic.

3.2 Code-mixed Pre-training

We train a multilingual model (Dabre et al., 2020; Firat et al., 2016; Johnson et al., 2017) model for synthetic Hinglish to English and English to synthetic Hinglish. We append a token indicating the source language at the end of the source sentence and a token indicating the target language at the beginning of the target sentence. This bidirectional model is trained till convergence on the development set provided by the organizers after the dev set evaluation phase. We expect that code-mixed pre-training, even if the Hinglish is synthetic, should help overcome the scarcity of code-mixed parallel corpus.

3.3 Multi-way Fine-tuning

We fine-tune the pre-trained model on Hinglish to English and English to Hinglish jointly. We use a small subset of the English side[1] of our synthetic data and the entire clean parallel corpus (PHINC+HinGE) together. We do this to prevent the model from overfitting on the small training data. The English subset is used as the source as well as the target and hence, in order to prevent the model from learning to copy the English data, we randomly mask spans of English tokens on the source. This is the same as denoising, which is used in BART (Lewis et al., 2020). This concept of using the pre-training data along with the fine-tuning data is also known as mixed fine-tuning (Dabre et al., 2019; Chu et al., 2017). As during pre-training, the development set data is used.

4 Experiments

We describe our experiments in our submissions.

4.1 Datasets and Pre-processing

We use the PHINC and HinGE datasets for our experiments. We do not use the synthetic parts of HinGE. During our preliminary experiments we used the development data provided with HinGE but found it to be unreliable and therefore used the development data provided after the first evaluation phase. We combined the data from both sources and overall we had 18,095 training instances for each direction for a total of 36,190 training instances. Note that HinGE has sources in English as well as Hindi, and this is also available for the development and test sets for translation into Hinglish. However, we do not explore multi-source translation in this paper. For pre-training, we used the Hindi–English part of the Samanantar dataset[2] (Ramesh et al., 2022) which contains 8.56M parallel sentences. We used the Romanization script from the Indic NLP Library[3] to convert

[1] We do not use the Hinglish side since it's synthetic and do not want it to interfere in the learning of actual Hinglish.

[2] https://indicnlp.ai4bharat.org/samanantar/

[3] https://github.com/anoopkunchukuttan/indic_nlp_library

Direction	Rouge-L	WER	Human Rating
Hinglish → English	0.52878 (4)	0.71517 (4)	2.85
English → Hinglish	0.46276 (4)	0.79271 (5)	1.00

Table 1: Official results of evaluation of Hinglish to English and English to Hinglish.

Devanagari to the Roman alphabet for Hindi. No other pre-processing was done.

4.2 Model Training and Decoding

We train transformer models (Vaswani et al., 2017) using the transformer-big settings. We used the YANMTT toolkit[4] (Dabre and Sumita, 2021) for training our models. We trained a joint Hinglish and English tokenizer of 16,000 subwords using all the synthetic and real training data we had. Pre-training was done on 8 NVIDIA V100 GPUs till convergence on the development data. (Mixed) Fine-tuning was done on a single GPU due to the relatively smaller size of the data. Once training has converged, we choose the checkpoints giving the highest development scores for decoding the test sets. We experimented with both BLEU and Rouge-L as metrics to determine convergence, but used BLEU as it is much stricter. We decode using beam search with a beam size of 32 and a length penalty of 1.6 both of which are empirically determined on the development set.

4.3 Results

Table 1 shows the official results obtained using the official evaluation servers. The organizers use Rouge-L and Word Error Rate (WER) as well as Human Ratings by evaluating 50 translations from our submissions. Overall, our automatic evaluation scores achieved a rank of 4 out of 8 participants. Compared to some of the baselines trained using only HinGE and PHINC, our main results using pre-training and fine-tuning are vastly better.

4.4 Analysis

We got a human rating score of 1 for translation into Hinglish and upon investigation we noted that our model simply copies the English sentence to the target. We are not sure why this happens. Regardless, on the development set, copying seems to give high BLEU and Rouge-L scores. However, the output is not Hinglish and is heavily penalized. We also did not conduct back-translation (Sennrich et al., 2016) of English into Hinglish due to this issue. We will probe our models deeper to understand why this happens. Due to lack of access to the official evaluation interface after the submission deadline, we were unable to conduct additional experiments.

5 Conclusion

In this paper, we have described our submission to the MixMT shared task at WMT 2022. We have used a combination of synthetic Hinglish–English parallel data creation, pre-training and fine-tuning to obtain our submissions which ranked 4th. Our analyses reveal that our English to Hinglish translation model actually ended up copying the English sentence as target. We will investigate and fix this in the future.

[4] https://github.com/prajdabre/yanmtt

References

Vibhav Agarwal, Pooja Rao, and Dinesh Babu Jayagopi. 2021. Hinglish to English machine translation using multilingual transformers. In *Proceedings of the Student Research Workshop Associated with RANLP 2021*, pages 16–21, Online. INCOMA Ltd.

Subhra Jyoti Baroi, Nivedita Singh, Ringki Das, and Thoudam Doren Singh. 2020. NITS-Hinglish-SentiMix at SemEval-2020 task 9: Sentiment analysis for code-mixed social media text using an ensemble model. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1298–1303, Barcelona (online). International Committee for Computational Linguistics.

Meghana Bhange and Nirant Kasliwal. 2020. HinglishNLP at SemEval-2020 task 9: Fine-tuned language models for Hinglish sentiment detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 934–939, Barcelona (online). International Committee for Computational Linguistics.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–391, Vancouver, Canada. Association for Computational Linguistics.

Alexis CONNEAU and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Comput. Surv.*, 53(5).

Raj Dabre, Atsushi Fujita, and Chenhui Chu. 2019. Exploiting multilingualism through multistage fine-tuning for low-resource neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1410–1416, Hong Kong, China. Association for Computational Linguistics.

Raj Dabre and Eiichiro Sumita. 2021. YANMTT: yet another neural machine translation toolkit. *CoRR*, abs/2108.11126.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2020. A semi-supervised approach to generate the code-mixed text using pre-trained encoder and transfer learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2267–2280, Online. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Puneet Mathur, Ramit Sawhney, Meghna Ayyar, and Rajiv Shah. 2018. Did you offend me? classification of offensive tweets in Hinglish language. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 138–148, Brussels, Belgium. Association for Computational Linguistics.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2022. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

Sebastin Santy, Anirudh Srinivasan, and Monojit Choudhury. 2021. BERTologiCoMix: How does code-mixing interact with multilingual BERT? In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 111–121, Kyiv, Ukraine. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Pranaydeep Singh and Els Lefever. 2020. Sentiment analysis for Hinglish code-mixed tweets by means of cross-lingual word embeddings. In *Proceedings of the The 4th Workshop on Computational Approaches to Code Switching*, pages 45–51, Marseille, France. European Language Resources Association.

Vivek Srivastava and Mayank Singh. 2020. PHINC: A parallel Hinglish social media code-mixed corpus for machine translation. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 41–49, Online. Association for Computational Linguistics.

Vivek Srivastava and Mayank Singh. 2021. HinGE: A dataset for generation and evaluation of code-mixed Hinglish text. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 200–208, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2021. Automatic bilingual markup transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3524–3533, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gui at MixMT 2022 : English-Hinglish : An MT approach for translation of code mixed data

Akshat Gahoi **Jayant Duneja** **Anshul Padhi** **Shivam Mangale**
Saransh Rajput **Tanvi Kamble** **Dipti Misra Sharma** **Vasudeva Varma**

International Institute of Information Technology, Hyderabad

{akshat.gahoi,anshul.padhi,saransh.rajput,tanvi.kamble}@research.iiit.ac.in
{dunejajayant,shivammangale}@gmail.com

Abstract

Code-mixed machine translation has become an important task in multilingual communities and extending the task of machine translation to code mixed data has become a common task for these languages. In the shared tasks of WMT 2022, we try to tackle the same for both English + Hindi to Hinglish and Hinglish to English. The first task dealt with both Roman and Devanagari script as we had monolingual data in both English and Hindi whereas the second task only had data in Roman script. To our knowledge, we achieved one of the top ROUGE-L and WER scores for the first task of Monolingual to Code-Mixed machine translation. In this paper, we discuss the use of mBART with some special pre-processing and post-processing (transliteration from Devanagari to Roman) for the first task in detail and the experiments that we performed for the second task of translating code-mixed Hinglish to monolingual English.

1 Introduction

Code Mixing occurs when a multi-lingual individual uses two or more languages while communicating with others. It is the most natural form of conversation for multilinguals. It is often confused with code-switching but there is a slight difference between the two. Both these phenomena include communicating in multiple languages but code switching usually takes place within multiple sentences while code mixing usually refers to words of different languages used in the same sentence. In code mixing, phrases, words and morphemes of one language may be embedded within an utterance of another language. Code mixing is extensively observed on social media sites like Facebook and twitter. With the rapid growth of social media and consequently, increase in the use of code-mixed data, it becomes important to develop systems to process such text.

Machine Translation, also known as automated translation, is the process where a software translates text from one language to another without any human involvement. There are multiple forms of machine translation, however, over the past few years, neural machine translation has become extremely popular. The WMT shared task had two subtasks. The first subtask consisted of the translation of Hindi-English parallel sentence pairs to Hindi-English code mixed sentences through machine translation. The second subtask consisted of the translation of Hindi-English code mixed sentences to English.

2 Background

While there is a growing interest in code-mixed text analysis as a research problem, there is one bottleneck that has hindered the growth of such works, and that is the lack of data. Due to this, there aren't many robust models for code-mixed text. To build standardized datasets of code-mixed text, we need to come up with ways of text generation of these code-mixed texts. These texts would be very helpful in training language models for various code-mixed pairs as language models only need unsupervised data.

Code Mixed text generation is a relatively new problem, and so is its initial stage. One of the recent works in this field (Rizvi et al., 2021) tried to use linguistic theories to synthetically build code-mixed text using parallel monolingual corpora of two languages. The Equivalence Constraint Theory (Poplack, 1980) says that code-mixing can only occur at parts of the text where the surface structures of two languages map onto each other. So in these parts, the grammatical rules of both languages are followed. The Matrix Language Theory (McClure, 1995) tries to solve this problem by separating the two languages into a base language and a second language. The grammatical rules of the base lan-

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1126 - 1130
December 7-8, 2022 ©2022 Association for Computational Linguistics

guage are followed, and parts of the base language are replaced by the corresponding parts of the second language whenever it is grammatically feasible to do so.

Deep Learning and Neural Networks have also been used to build systems for code mixed generation. In these systems, the problem of text generation has been posed as one of machine translation, where monolingual text is translated to code-mixed text. Some of the early work involved using the then state of the art encoder-decoder models like pointer generator networks(Winata et al., 2019) and GANs(Chang et al., 2019) to translate two sets of monolingual corpora into code mixed text. With the rise of multilingual models like mT5 (Xue et al., 2020), mBART, indicBART (Dabre et al., 2021), etc. the task of translation has become much easier as these models understand both languages and this has been shown to outperform previous models in many workshops.

mBART(Liu et al., 2020) is a denoising autoencoder which has been trained on a very large dataset which contains text from 25 languages. It has the same transformer based architecture and training objective as BART, a denoising autoencoder which was shown to be one of the best performing sequence to sequence models at the time. It has been trained to reconstruct original text which has been corrupted as a way to add noise. It can perform various downstream sequence to sequence tasks like machine translation, text summarization, etc. mBART consists of 12 encoder layers and 12 decoder layers. There are 16 heads and a model dimension of 1024.

Another solution to circumvent the data problem is to create translation systems that can translate code-mixed text to monolingual text. This allows us to use robust NLP systems for various downstream tasks.

While we have the above said top performing models at the moment, they are very heavy computational wise due to their large parameter sizes. With resource constraints, it is tough to replicate their performance. Helsinki's OPUS-MT (Tiedemann and Thottingal, 2020) model was of comparabaly smaller size and focused on the initiative of supporting low-resource languages. It does accordingly have lower performance. We have attempted at utilizing this model in our case with further training on provided data to understand whether under the resource constraints, we can observe competitive performance.

Data	Length
Synthetic (Train)	3263
Synthetic (Validate)	396
Human Generated (Train)	1800
Human Generated (Validate)	376

Table 1: Distribution of Sentences in the data

The model architecture is based on a standard transformer setup with 6 self-attentive layers in the encoder and decoder network. It has 8 attention heads in each layer. This is hence comparatively low compute seeking as compared to the mainstream models.

3 System Overview

3.1 Task 1

In this section we propose our system for Task 1 which is English and Hindi to code-mixed text translation

3.1.1 Dataset and Data Preparation

The dataset that we used for Task 1 was the HinGE dataset (Srivastava and Singh, 2021). It is divide into two parts, the synthetic dataset or the machine generated dataset and the human generated dataset. (Table 1) There were 3659 and 2176 sentences respectively.

3.1.2 Model

In this task we finetune the mBART model on the data given to us. Since mBART is a very large model we needed to decrease its size. We do this by reducing the vocabulary of the model as the vocabulary adds to the model size by a lot and we don't need the vocabulary from the rest of the 25 languages. To reduce the vocabulary we create our own vocabulary using the tokens present in the task dataset, IIT-B English-Hindi parallel corpus (Kunchukuttan et al., 2018) and the Dakshina Dataset (Roark et al., 2020) as we feel the two datasets were large enough to create a vocabulary extensive enough to solve the given task. We process the input data from the given task data as explained above to create our input. Using the corresponding code-mixed sentences as the gold output we finetune the mBART model.

3.1.3 Post Processing

The output of our model was in a mixed script (Roman + Devanagari). So the post processing

Post Processing	ROUGE-L	WER
Normal Output	0.39091	0.81884
With Automated Transliteration	0.48376	0.72561
With Automated Transliteration + Dictionary Based Transliteration	**0.61667**	**0.63342**

Table 2: ROUGE-L and WER scores after different post processing tasks

becomes one of the important step in this task as we wanted our output Hinglish sentences to be only in Roman script. We used transliteration function from indicate library as our first step to see how good the results will be. There were many instances where the transliteration done by indicate was not accurate. So the next step that we did was to create a dictionary of most common words and numbers with their corresponding transliterated Roman text. This dictionary over the automatic transliteration by indicate was used to get the best output of our model in the Roman script.

3.2 Task 2

In this section we propose our system for Task 2 which is Hinglish (code-mixed) to English text translation.

3.2.1 Observations

The data for task 2 are tweets based data. Due to the tweets nature, we observed that:

- The URLs included tended to be at the end of the sentences.
- The mentions (of the form '@<some_user_tag>' for instance @LokSabha) at the beginning and the end of tweets are generally such that the sentences can be translated without them with no-low loss of information.
- Hashtags which are added at the end of the tweets are generally for increasing outreach and exposure.

Based on the above observations, we found that the information provided by these tokens to the translation was not significant as compared to the loss of information due to incorrect translation of these units. Hence, we applied heuristics to appropriately preprocess the input data to exclusively and exhaustively split the tweets into sentences (which will be translated), URLs, mentions and hashtags, which are then concatenated after the translation in postprocessing.

3.2.2 Dataset and Data Preparation

The dataset that we used for Task 2 was the PHINC dataset (Srivastava and Singh, 2020). It contains 13,738 parallel sentences in Hinglish (code-mixed) and English of which we used a train-val-test split of 80-10-10. We transliterated the Hinglish sentences from the Roman script to the Devanagiri script using the Google Transliterate API, to utilise pre-trained Hindi to English translation models. This transliterator was used among others due to it having one of the best performance, it's similarity in the vocabulary space with the input dataset as compared to the other transliterators available and also that PHINC was jointly created using Google Translate.

3.2.3 Model

Due to compute constraints, we decided to utilize pretrained models, that would be efficient for our dataset. To access better models, we went ahead with models trained with a task or a subtask of Hindi to English machine translation. We appropriately processed the data for the same. We hence decided to finetune Salesken.AI's pretrained model provided on Huggingface Transformers. They have finetuned Helsinki's OPUS-MT model on AI4Bharat's Samanantar dataset (Ramesh et al., 2021), a large indic dataset.

4 Experimental Setup

In task 1, we use the fairseq implementation of mBART as our base model which has been trained on 4 Nvidia GeForce RTX 2080 Ti GPUs. The model has been trained using label smoothed cross entropy as the loss criterion. The model uses an Adam optimizer with polynomial decay learning rate scheduling, dropout = 0.3, learning rate = $3 * 10^{-5}$, $\epsilon = 10^{-6}$, $\beta_1 = 0.9$ and $\beta_2 = 0.98$.

The model was trained on 10000 steps with 2500 warm up steps and a batch size of 512 tokens.

We validate the model on each epoch on a validation set and at the end we select the model with the lowest loss.

In task2, for fine-tuning we use the Salesken.AI's pretrained model provided on Huggingface Transformers. The model was trained on Nvidia GeForce RTX 2080 Ti GPUs.The model has been trained using label smoothed cross entropy as the loss criterion. The model uses an Adam optimizer with learning rate = $3 * 10^{-4}$, $\epsilon = 10^{-9}$, β_1 = 0.9 and β_2 = 0.98.

5 Results and Evaluation

The test dataset consisted of 500 sentences. These sentences also had both English sentence and its corresponding Hindi sentence. ROUGE-L score and WER score was considered for evaluation. ROUGE-L score considers longest common subsequence for its scoring. It counts the longest subsequence which is shared between both reference and the output. Its different from precision as it only counts the ratio between longest subsequence matched and the number of words matched. It does not take all the words in the reference.

The WER score represents the word error rate. Total errors between the reference and output is considered for this score. It adds up all the substitution, addition and deletion required to convert the output to the reference sentence and treat it as total error of the output. It can be treated same as calculating Levenshtein distance.

So our aim was to maximise ROUGE-L score and minimize WER score. Our score improved as we translitered the output from Devanagari to Roman using indicate library. The score increase significantly after we created a dictionary of words for transliterating most common Hindi words and numbers. We achieved a ROUGE-L score of 0.61667 and WER score of 0.63342 after both the post processing steps.

The test set provided for Task 2 contained 1500 lines, which were processed as mentioned in 5 The results for the evaluation metrics we obtained for the test set provided for Task 2 is available in 3. Using the Google Transliterate API significantly improved the quality of the input data, and also the similarity of vocabulary with the dataset as mentioned earlier. The application of heuristics also bolstered the approach's performance.

Based on qualitative evaluation, it was observed that it struggled to get long sentence translations which can be attributed to the source of the dataset being of of tweets which have a noisy and inconsistent structure. This is alongside the lower parameter size and attention heads.

Metric	Score
ROUGE-L	0.41493
WER	0.80804

Table 3: Results for Task 2

The model was trained till significant learning on a wide array of parameters, till resource permits, in an attempt to provide more opportunities to appropriately fine-tune the model, but even though there was a sign of the model learning, the performance was observed to be not competitive to the current top performers.

6 Conclusion

In this paper, we approached code mixed machine translation problem from both the direction. We used mBART for our first task of translating English and corresponding Hindi sentences to Hinglish sentence. The results were significantly improved through transliterating the output from Devanagari script to Roman script. Two different methods were used for the same. Our model surpassed baseline in ROUGE-L and WER scores by a huge margin.

For the second task of translating Hinglish sentences to English sentence by fine-tuning Salesken.AI's pre trained model. We cleared the baseline but their is still work to be done in that field as we think that it can be further improved. For the future work in this area we would like to work further on the second task in hand of translating codemixed language to a monolingual language. We need to retrieve information about both the languages from the code mixed sentence and try to give a output in a mono lingual langauge without disturbing the word order.

References

Ching-Ting Chang, Shun-Po Chuang, and Hung-Yi Lee. 2019. Code-Switching Sentence Generation by Generative Adversarial Networks and its Application to Data Augmentation. In *Proc. Interspeech 2019*, pages 554–558.

Raj Dabre, Himani Shrotriya, Anoop Kunchukuttan, Ratish Puduppully, Mitesh M. Khapra, and Pratyush Kumar. 2021. Indicbart: A pre-trained model for natural language generation of indic languages.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation.

Erica McClure. 1995. Duelling languages: Grammatical structure in codeswitching. carol myers-scotton. oxford: Clarendon press, 1993. pp. xiv 263. *Studies in Second Language Acquisition*, 17(1):117–118.

Shana Poplack. 1980. Sometimes i'll start a sentence in spanish y termino en espaÑol: toward a typology of code-switching 1. *Linguistics*, 18:581–618.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2021. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages.

Mohd Sanad Zaki Rizvi, Anirudh Srinivasan, Tanuja Ganu, Monojit Choudhury, and Sunayana Sitaram. 2021. GCM: A toolkit for generating synthetic code-mixed text. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 205–211, Online. Association for Computational Linguistics.

Brian Roark, Lawrence Wolf-Sonkin, Christo Kirov, Sabrina J. Mielke, Cibu Johny, Isin Demirsahin, and Keith Hall. 2020. Processing south asian languages written in the latin script: the dakshina dataset.

Vivek Srivastava and Mayank Singh. 2020. PHINC: A parallel Hinglish social media code-mixed corpus for machine translation. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 41–49, Online. Association for Computational Linguistics.

Vivek Srivastava and Mayank Singh. 2021. Hinge: A dataset for generation and evaluation of code-mixed hinglish text. pages 200–208.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2019. Code-switched language models using neural based synthetic data from parallel sentences. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 271–280, Hong Kong, China. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer.

MUCS@MixMT: indicTrans-based Machine Translation for Hinglish Text

Asha Hegde[a], **Hosahalli Lakshmaiah Shashirekha**[b]
Department of Computer Science, Mangalore University, Mangalore, India
{[a]hegdekasha,[b]hlsrekha}@gmail.com

Abstract

Code-mixing is the phenomena of mixing various linguistic units such as paragraphs, sentences, phrases, words, etc., of two or more languages in any text. It is predominantly used to post the comments by social media users who know more than one language. Processing code-mixed text is challenging because of its complex characteristics and lack of tools that support such data. Developing efficient Machine Translation (MT) systems for code-mixed text is challenging due to lack of code-mixed data. Further, existing MT systems developed to translate monolingual data are not portable to translate code-mixed text mainly due to the informal nature of code-mixed data. To address the MT challenges of code-mixed text, this paper describes the proposed MT models submitted by our team MUCS, to the Code-mixed Machine Translation (MixMT) shared task in the Workshop on Machine Translation (WMT) organized in connection with Empirical models in Natural Language Processing (EMNLP) 2022. This shared task has two subtasks: i) subtask 1 - to translate English sentences and their corresponding Hindi translations written in Devanagari script into Hinglish (English-Hindi code-mixed text written in Latin script) text and ii) subtask 2 - to translate Hinglish text into English text. The proposed models that translate English text to Hinglish text and vice versa, comprise of i) transliterating Hinglish text from Latin to Devanagari script and vice versa, ii) pseudo translation generation using existing models, and iii) efficient target generation by combining the pseudo translations along with the training data provided by the shared task organizers. The proposed models obtained 5th and 3rd rank with Recall-Oriented Under-study for Gisting Evaluation (ROUGE) scores of 0.35806 and 0.55453 for subtask 1 and subtask 2 respectively.

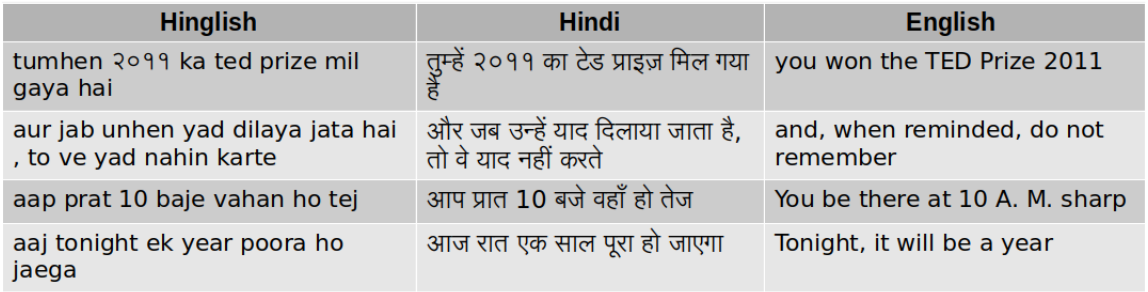
1 Introduction

In linguistic terms, code-mixing is the practice of switching between two or more languages within or across sentences/words in any text (Joshi, 1982). Due to the widespread use of social media platforms like Twitter, Facebook, Reddit, etc., users are generating more and more code-mixed content. In Indian scenario, social media users usually blend English with their mother tongue or local language, for instance, English and Hindi, mainly for the technological limitations of computer keyboard or smartphone keypads to enter text in local languages. Further, as most of the text processing tasks are developed for handling monolingual and formal text, informal and/or code-mixed text such as Hinglish is less explored. As the code-mixed text like Hinglish is increasing day by day, many applications such as MT, sentiment analysis, emotion analysis, etc., are also increasing. This has created a great demand for the tools and resources to process code-mixed data. Sample Hinglish text along with their Hindi and English translations are given in Table 1.

In recent years, pre-trained transformer-based language models have become state-of-the-art models for most of the downstream tasks including MT, text classification, text generation, and natural language understanding. To train such models, underlying data is drawn from a sizable monolingual corpus that is available in Wikipedia, book corpora, etc. Several models like Multilingual Bidirectional and Auto-Regressive Transformer (mBART) (Liu et al., 2020) and Multilingual Text to Text Transformer (mT5) (Xue et al., 2021) are readily available for many languages. However, due to the scarcity of code-mixed corpus, developing the pretrained language models for code-mixed text is very challenging.

MT being one of the important applications of code-mixed texts mainly focuses on translating monolingual text leaving aside the code-mixed data. Further, for under-resourced languages with rich morphological features like Hindi (Sangwan and Bhatia, 2021), developing MT models become more challenging in the code-mixed sce-

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1131 - 1135
December 7-8, 2022 ©2022 Association for Computational Linguistics



Hinglish	Hindi	English
tumhen २०११ ka ted prize mil gaya hai	तुम्हें २०११ का टेड प्राइज़ मिल गया है	you won the TED Prize 2011
aur jab unhen yad dilaya jata hai , to ve yad nahin karte	और जब उन्हें याद दिलाया जाता है, तो वे याद नहीं करते	and, when reminded, do not remember
aap prat 10 baje vahan ho tej	आप प्रात 10 बजे वहाँ हो तेज	You be there at 10 A. M. sharp
aaj tonight ek year poora ho jaega	आज रात एक साल पूरा हो जाएगा	Tonight, it will be a year

Table 1: Sample Hinglish text and their Hindi and English translations

nario. To address these challenges, in this paper, we - team MUCS, describe the models submitted to MixMT-2022[1] shared task organized by WMT-2022 at EMNLP 2022. The shared task consists of two subtasks: i) subtask 1 - to translate English sentences and their corresponding Hindi translations into Hinglish text and ii) subtask 2 - to translate Hinglish text into English text. The proposed methodology consists of i) transliteration of Hinglish text from Latin script to Devanagari script and vice versa, ii) generating pseudo translations for monolingual data using pretrained MT models, and iii) target generation by fine-tuning the pretrained models with a combination of the pseudo parallel data obtained as the output of pseudo translations and the dataset provided by the organizers of the shared task.

The following is a breakdown of the paper's structure: Section 2 contains the related work and the proposed methodology is explained in Section 3. Section 4 gives the details about experiments and results and the paper concludes in Section 5 with future work.

2 Related work

Due to the increasing amount of code-mixed text, MT of code-mixed text is gaining attention of the researchers and the description of few of the models developed to translate Hinglish text into English text and vice versa are given below:

Srivastava and Singh (2020) manually developed a parallel corpus of 13,738 Hinglish sentences and their translations in English with the help of 54 annotators. They proposed a simple tagging approach for tagging each token in a sentence with the language it belongs to and evaluated Bing Translate (BT) and Google Translate (GT) models - the two popular MT services using their parallel corpus. Among the two models, GT model outperformed with a better Bilingual Evaluation Understudy (BLEU) score of 0.153 when compared to that of BT. Dhar et al. (2018) manually developed a Hinglish-English parallel corpus of 6,096 parallel sentences with the help of 4 human translators. Using a language identification technique, they tagged every word in a sentence with the name of the language to which it belongs. They proposed an MT model comprising three steps: i) identifying the matrix language, ii) translation of source text into matrix language, and iii) translation of matrix language into the target language by training BT. Considering steps i) and ii) as preprocessing, they obtained considerable translation with BLEU score of 25.0.

Jawahar et al. (2021) created a parallel corpus of 17.8 million English-Hinglish sentence pairs by leveraging bilingual word embeddings to translate English text into Hinglish text and vice versa. Further, they fine-tuned mBART and mT5 - the pretrained text generators using their newly constructed parallel corpus. The mT5 model obtained a better BLEU score of 13.95 compared to that of mBART. Gautam et al. (2021) proposed an effective fine-tuning of mBART using English-Hinglish dataset[2] to translate English text to Hinglish text and vice versa. They transliterated the Hinglish text in Latin script to Devanagari script to fine-tune the mBART model and obtained BLEU scores of 11.86 and 12.22 for Hinglish to English and English to Hinglish translations respectively.

From the literature, it is clear that very few attempts are made to explore English-Hinglish code-mixed parallel corpus for MT. Hence, there is enough space to explore new techniques in this direction.

[1] https://codalab.lisn.upsaclay.fr/competitions/2861#learn_the_details

[2] https://code-switching.github.io/2021

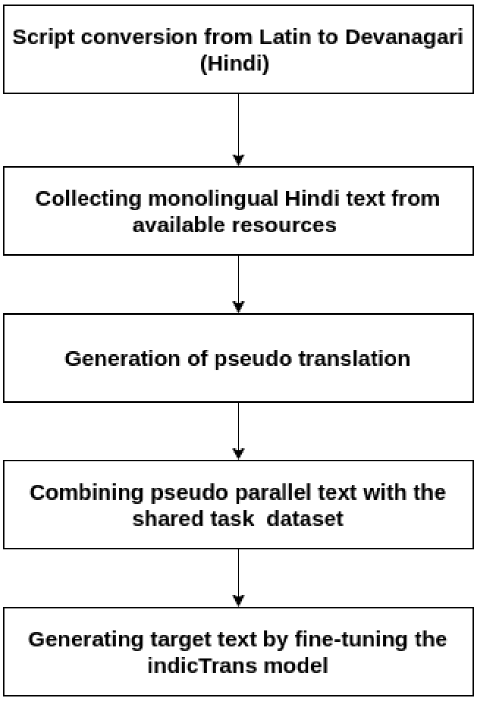

Script conversion from Latin to Devanagari (Hindi)
Collecting monolingual Hindi text from available resources
Generation of pseudo translation
Combining pseudo parallel text with the shared task dataset
Generating target text by fine-tuning the indicTrans model

Figure 1: Workflow of the proposed method

3 Proposed methodology

Inspired by Gautam et al. (2021), a pipeline of transliteration and fine-tuning indicTrans[3] (used for translation) pretrained models is proposed to address subtask 1 and subtask 2. EN-Indic and Indic-EN models are used for generating Hinglish text and pseudo translations respectively. The purpose of transliteration is to utilize pretrained models which are trained using the text in their native script. Further, the proposed methodology also consists of the generation of pseudo translations where pseudo translation mimics the translation process.

The framework of the proposed model is given in Figure 1 and the system descriptions of each subtasks are given below:

Subtask 1 - In addition to the dataset provided by the organizers[4] for this shared task, monolingual Hindi text is collected from the available resources[5] and the further steps used to accomplish the subtask 1 are given below:

1. Transliteration is carried out using indic-trans[6] to transliterate Hinglish text in Latin script to Devanagari script
2. EN-Indic and Indic-EN models trained on Samanantar[7] corpus are used for translations (Ramesh et al., 2022)
3. Pseudo translations are generated using the Indic-EN model considering monolingual Hindi text
4. The shared task dataset is combined with the pseudo parallel data and EN-Indic model is then fine-tuned on this data
5. Finally, the target Hinglish text is generated by transliterating Devanagari script to Latin script using indic-trans

Subtask 2 - For subtask 2, the procedure similar to that of subtask 1 is followed considering Hinglish text as the source and English text as the target to generate the required output.

Subtask	Train set	Development set	Test set
subtask 1	2,766	500	1,500
subtask 2	13,738	500	1,500

Table 2: Statistics of the shared task dataset for both the subtasks in terms of the number of sentences

4 Experiments and Results

The statistics of the dataset provided by the organizers for both the subtasks which are used to build the proposed models are given in Table 2. The data provided for subtask 1 is the synthetic data (Srivastava and Singh, 2021) which consists of English sentences and their corresponding Hindi translations as the source and Hinglish as the target. For subtask 2, the dataset consists of Hinglish-English sentence pairs (Srivastava and Singh, 2020) to generate English text.

EN-Indic and Indic-EN models which are trained on Samanantar corpus are fine-tuned with the combination of the shared task dataset and pseudo parallel text. Exhaustive experiments are carried out to get the best results by tuning the hyperparameters, which control the learning process of EN-Indic and Indic-En models. Table 3 gives the hyperparameters and their values used to fine-tune the EN-Indic and Indic-EN models that gave the best results on development set.

The user predictions for the given Test set submitted to the organizers of the shared task are evaluated based on ROUGE score and Word Error Rate (WER). ROUGE score is calculated based

[3] https://indicnlp.ai4bharat.org/indic-trans/
[4] Codalab competitions
[5] https://indicnlp.ai4bharat.org/samanantar/
[6] https://github.com/libindic/indic-trans
[7] https://indicnlp.ai4bharat.org/samanantar/

Hyperparameters	Values
max-token	1,568
learning rate	0.00003
label smoothing	0.1
optimizer	adam
dropout	0.2

Table 3: Hyperparameters and their values used to fine-tune the EN-Indic and Indic-EN models

Subtask		ROUGE	WER
subtask 1	Development set	0.38935	0.72310
	Test set	0.35806	0.76096
subtask 2	Development set	0.54556	0.65938
	Test set	0.55453	0.64737

Table 4: Performance measures of the proposed method for both Development set and Test set

on the overlapping of n-grams between the candidate string and reference string whereas WER score is calculated by dividing the number of errors by the total number of words. Performance measures of the proposed models for both Development set and Test set is given in Table 4. From Table 4, it is clear that the ROUGE score of subtask 2 is better than subtask 1 as the dataset used for subtask 1 is very small compared to that of subtask 2. Further, the performance of Indic-EN model is better than EN-Indic model (Ramesh et al., 2022) and the same is reflected in Table 4. The comparison of the ROUGE score of the proposed models with the models submitted by all the participants of the shared task for subtask 1 and subtask 2 are shown in Figure 2 and 3 respectively. From Figure 2 and 3, it is clear that the proposed method obtained considerable ROUGE scores for both the subtasks.

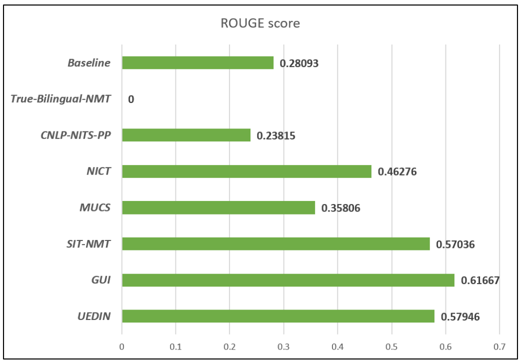


Figure 2: Comparison of ROUGE score of participated teams with the proposed model for subtask 1

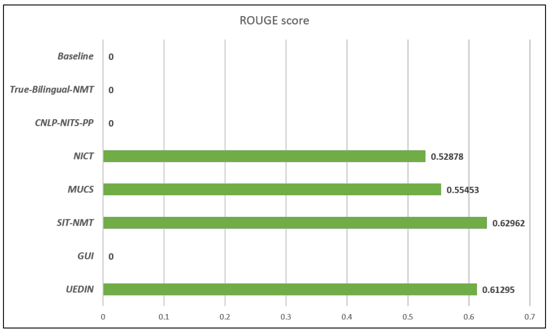


Figure 3: Comparison of ROUGE score of participated teams with the proposed model for subtask 2

5 Conclusion and Future work

This paper describes the models submitted by our team - MUCS to MixMT 2022 shared task to perform MT from English text and their corresponding Hindi translations into Hinglish text and from Hinglish text to English. The proposed models consist of transliteration and pseudo translation generation followed by fine-tuning the pretrained MT models using the combination of pseudo parallel data and the shared task dataset for target generation. These models obtained ROUGE scores of 0.35806 and 0.55453 securing 5th and 3rd rank for subtask 1 and subtask 2 respectively. The efficient transliteration techniques with effective fine-tuning of the pretrained models for code-mixed Hinglish translation will be explored further.

References

Mrinal Dhar, Vaibhav Kumar, and Manish Shrivastava. 2018. Enabling Code-Mixed Translation: Parallel Corpus Creation and MT Augmentation Approach. In *Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing*, pages 131–140.

Devansh Gautam, Prashant Kodali, Kshitij Gupta, Anmol Goel, Manish Shrivastava, and Ponnurangam Kumaraguru. 2021. CoMeT: Towards Code-Mixed Translation Using Parallel Monolingual Sentences. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 47–55.

Ganesh Jawahar, El Moatez Billah Nagoudi, Muhammad Abdul-Mageed, and Laks VS Lakshmanan. 2021. Exploring Text-to-Text Transformers for English to Hinglish Machine Translation with Synthetic Code-mixing. In *arXiv preprint arXiv:2105.08807*.

Aravind K. Joshi. 1982. Processing of Sentences With Intra-Sentential Code-Switching. In *Coling 1982: Proceedings of the Ninth International Conference on Computational Linguistics*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual Denoising Pretraining for Neural Machine Translation. In *Transactions of the Association for Computational Linguistics*, pages 726–742.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Divyanshu Kakwani, Navneet Kumar, et al. 2022. Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages. In *Transactions of the Association for Computational Linguistics*, pages 145–162. MIT Press.

Saurabh R Sangwan and MPS Bhatia. 2021. Denigrate Comment Detection in Low-resource Hindi Language using Attention-based Residual Networks. In *Transactions on Asian and Low-Resource Language Information Processing*, pages 1–14.

Vivek Srivastava and Mayank Singh. 2020. PHINC: A parallel Hinglish Social Media Code-mixed Corpus for Machine Translation. In *arXiv preprint arXiv:2004.09447*.

Vivek Srivastava and Mayank Singh. 2021. Hinge: A Dataset for Generation and Evaluation of Code-mixed Hinglish Text. In *arXiv preprint arXiv:2107.03760*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.

SIT at MixMT 2022:
Fluent Translation Built on Giant Pre-trained Models

Abdul Rafae Khan, Hrishikesh Kanade, Girish Amar Budhrani,
Preet Jhanglani, and **Jia Xu**
Stevens Institute of Technology
{akhan4, hkanade, gbudhran, pjhangl1, jxu70}@stevens.edu

Abstract

This paper describes the Stevens Institute of Technology's submission for the WMT 2022 Shared Task: Code-mixed Machine Translation (MixMT). The task consisted of two subtasks, subtask 1 Hindi/English to Hinglish and subtask 2 Hinglish to English translation. Our findings lie in the improvements made through the use of large pre-trained multilingual NMT models and in-domain datasets, as well as back-translation and ensemble techniques. The translation output is automatically evaluated against the reference translations using ROUGE-L and WER. Our system achieves the 1^{st} position on subtask 2 according to ROUGE-L, WER, and human evaluation, 1^{st} position on subtask 1 according to WER and human evaluation, and 3^{rd} position on subtask 1 with respect to ROUGE-L metric.

1 Introduction

Code-mixing (or code-switching) is the phenomenon when another language like Hindi is interleaved with English words in the same sentence. This code-mixed language is mostly used in social media text and is colloquially referred to as Hinglish. Despite Hindi being the fourth most widely spoken language in the world (Lewis, 2009), research in Hinglish translation has been a relatively unexplored task.

A major challenge in creating a translation system for code-mixed text is the limited amount of parallel data (Ranathunga et al., 2021). Typical methods use standard back-translation techniques (Sennrich et al., 2015a) for generating synthetic parallel data for training. Massive multilingual neural machine translation (NMT) models have recently been shown to improve the translation performances for low-resource and even zero-shot settings. We propose using such large multilingual NMT models for our code-mixed translation tasks.

Previous work has only used smaller multilingual architectures (Gautam et al., 2021). We use pre-trained multilingual models trained in up to 200 language directions. We finetune these models for the Hindi to Hinglish and Hinglish to English tasks. One major challenge when using these massive models is the GPU memory constraint. Another issue is the ratio of English and Hinglish words interleaved for each translation output. We use multiple state-of-the-art GPUs with model parallelization to overcome the memory issue. For the amount of English in the outputs, we tune the model parameters including learning rate, dropout, and the number of epochs to get the optimal translations.

Along with these pre-trained multilingual NMT models, we also use additional in-domain data, back-translation to generate additional parallel data, and using multi-run ensemble to improve the final performance. All these methods gave us an improvement of +24.4 BLEU for Hindi to Hinglish translation (subtask 1) and +28.1 BLEU points for Hinglish to English translation (subtask 2) compared to using only the organizer provided data and the baseline experiment.

In this paper, we discuss our submission for the WMT 2022 MixMT shared task. We participate in both the subtasks and our submission system includes the following:

- Tune very large pre-trained multilingual NMT models and finetune on in-domain datasets;
- Back-translation to create synthetic data for in-domain monolingual data;
- Multi-run ensemble to combine models trained on various datasets;

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1136 - 1144
December 7-8, 2022 ©2022 Association for Computational Linguistics

- Tune model parameters to enhance model performance.

2 Related Work

Multilingual Neural Machine Translation (MNMT) Word and subword-level tokenizations are widely used in natural language processing, including NMT/MNMT. Morishita et al. (2018) propose to incorporate hierarchical subword features to improve neural machine translation. Massively multilingual NMT models are proposed by Aharoni et al. (2019) and Arivazhagan et al. (2019). They are trained on a large number of language pairs and show a strong and positive impact on low-resource languages. However, these models tend to have representation bottlenecks (Dabre et al., 2020), due to the large vocabulary size and the large diversity of training languages. Two MNMT systems (Tan et al., 2019; Xiong et al., 2021) are proposed to solve this problem by modifying the model architectures, adding special constraints on training, or designing more complicated preprocessing methods. Xiong et al. (2021) adopt the contrastive learning scheme in many-to-many MNMT. Tan et al. (2019) propose a distillation-based approach to boost the accuracy of MNMT systems. However, these word/subword-based models still need complex preprocessing steps such as data augmentation or special model architecture design.

Code-mixed NMT The majority of research for code-mixed translation focuses on generating additional data using back-translation methods. Winata et al. (2019) used the sequence to sequence models to generate such data and Garg et al. (2018) used a recurrent neural network along with a sequence generative adversarial network. Pratapa et al. (2018) generated linguistically-motivated sequences. Additionally, there have been several code-mixed workshops (Bhat et al., 2017; Aguilar et al., 2018) to advance the field of code-mixed data.

Hindi or Hinglish NMT Researchers have worked on machine translation from Hindi to English (Laskar et al., 2019; Goyal and Sharma, 2019), however, there has been far less work for Hinglish. A major issue is the lack of parallel Hinglish-English data. Additional parallel data generated by back-translation is used to improve the performance (Gautam et al., 2021; Jawahar et al., 2021). The CALCS'21 competition (Solorio et al., 2021) had a shared task for English to Hinglish for movie review data.

3 Background

3.1 Task Description

The WMT 2022 CodeMix MT task consists of two subtasks. Subtask 1 is to use Hindi or English as input and automatically translate it into Hinglish. Subtask 2 is to input a Hinglish text and translate it into English. Participation in both subtasks was compulsory for the competition. We use Hindi only as the source for subtask 1.

3.2 Neural Machine Translation

The Neural Machine Translation (NMT) task uses a neural network-based model to translate a sequence of tokens from one human language to another. More formally, given a sequence of tokens in source language $x = \{x_1, x_2, \cdots, x_n\}$, the model outputs another sequence of tokens in target language $y = \{y_1, y_2, \cdots, y_m\}$. The input sequence x is encoded into the latent representation by a neural network-based encoder module, and these representations are decoded by the neural network-based decoder module. We train transformer-based encoder-decoder models (Vaswani et al., 2017) to translate the data. These models use a self-attention mechanism in their architectures to boost performance.

3.3 Multilingual NMT (MNMT)

Initial NMT systems were only capable of handling two languages. However, lately, there has been a focus on NMT models which can handle input from more than two languages (Dong et al., 2015; Firat et al., 2016; Johnson et al., 2017). Such models, commonly called Multilingual NMT (MNMT) models, have shown improvement in low-resource or zero-shot Neural Machine Translation settings. Instead of translating a sequence of tokens in source language x to another sequence in target language y, the MNMT system uses multiple sources and target languages.

There are two main approaches: (1) use a separate encoder and decoder for each of the source and target languages (Gu et al., 2018), and (2) use a single encoder/decoder which shares the parameters across the different languages (Johnson et al., 2017).

The issue with the first approach is that it requires a much larger memory due to multiple encoders and decoders (Vázquez et al., 2018). The second approach is much more memory efficient due to parameter sharing (Arivazhagan et al., 2019).

Training a model using the second approach can be done by adding a language tag to the source and target sequence. Specifically, when the decoding starts, an initial target language tag is given as input, which forces the model to output in that specific language.

4 Methods

For the initial set of experiments, we use the baseline transformer model (Vaswani et al., 2017). For all the other experiments, we use pre-trained multilingual NMT models and fine-tuned them for the specific datasets. We can divide these into three groups based on the number of parameters. (1) smaller models including mBART-50 (Tang et al., 2020) and Facebook M2M-100 medium model (Fan et al., 2021) (M2M-100), (2) the medium models include the Facebook NLLB-200 (Costa-jussà et al., 2022) (NLLB-200) and Google mT5 XL (Xue et al., 2021) (mT5-XL), and (3) for large model we use the Google mT5 XXL model (Xue et al., 2021) (mT5-XXL). The parameter count for each of the models and the training time per epoch for baseline datasets are mentioned in Table 1.

For both subtasks, we use Hindi as the source language tag and English as the target language tag.

4.1 Pre-trained Models

To train the transformer, mBART-50, and M2M-100 models, we use the Fairseq toolkit (Ott et al., 2019), and the larger NLLB-200, mT5-XL, and mT5-XXL models use the Huggingface toolkit (Wolf et al., 2019). Table 1 lists the parameter count for each pre-trained multilingual model.

Model	Params
mBART-50	611M
M2M-100	1.2B
NLLB-200	3.3B
mT5-XL	3.7B
mT5-XXL	13B

Table 1: Parameter count for each pre-trained multilingual model.

4.2 Data Augmentation

We use three different ways to add additional in-domain data for training our models.

Additional in-domain data We use additional in-domain parallel data and add it to the training data for accuracy improvement. Since our focus is on Hindi for subtask 1 and Hinglish for subtask 2, we tried to look for data from additional domains with Hindi or Hinglish as the source. We use Kaggle Hi-En (Chokhra, 2020) and MUSE Hi-En dictionary (Lample et al., 2017) for subtask 1. For subtask 2, we use Kaggle Hg-En data (Tom, 2022), CMU movie reviews data (Zhou et al., 2018), and CALCS'21 Hg-En dataset (Solorio et al., 2021). We also use selected WMT'14 News Hi-En sentences (Bojar et al., 2014) and the MTNT Fr-En and Ja-En data (Michel and Neubig, 2018). Table 2 all lists these datasets.

Back-translation A common technique used to increase the data size for low-resource languages is to use in-domain monolingual data and generate synthetic translations using a reverse translation system (Sennrich et al., 2015a). We use google translate for back-translation. We translate samples from the English side of Tatoeba Spanish to the English dataset (Tatoeba, 2022) and Sentiment140 dataset (Go et al., 2009) into Hinglish and use the synthetic translations as additional bilingual data.

4.3 Ensemble

We use a multi-run ensemble (Koehn, 2020) to combine multiple model's best checkpoints to boost the final performance. We average the probability distribution over the vocabulary for all the models to generate a final probability distribution and use that to predict the target sequence.

Dataset	Sentences	V_R	V
HinGE Hi-Hg	2.3K	103K	19K
PHINC Hg-En	13K	302K	55K
HinGE Hg-En	11K	109K	22K
Kaggle Hi-En	11K	220K	31K
Kaggle En-Hg	1.8K	98K	17K
MUSE Hi-En	30K	29K	24K
CMU Reviews Hg-En	8K	180K	24K
CALCS'21 Hg-En	8K	182K	23K
Back-translation Hg-En	8.5K	48K	7K
WMT'14 Hi-En	15K	181K	21K
MTNT Fr-En	10K	16K	14K
MTNT Ja-En	3.5K	120k	8K

Table 2: Datasets provided by the organizers and additional in-domain and out-of-domain datasets used for subtask 1 and 2. V_R is the number of running words and V is the vocabulary size.

5 Datasets

The competition provided one dataset for each of the subtasks, HinGE Hi-Hg (Srivastava and Singh, 2021) for subtask 1 and PHINC Hg-En (Srivastava and Singh, 2020) for subtask 2. The competition also provided the validation data. In addition to these, we also use additional in-domain and out-of-domain datasets.

Due to a large overlap of English and Hinglish vocabulary, we use Hindi-English (Hi-En) and Hindi-Hinglish (Hi-Hg) datasets for subtask 1. For subtask 2, we use various Hinglish-English datasets. All the competition provided datasets, the additional in-domain datasets, and the additional out-of-domain datasets used for both the subtasks are listed in Table 2. As HinGE En-Hg has multiple Hinglish translations for a single English sentence. We duplicated the English to increase the size of the data. For the WMT'14 Hi-En dataset, we selected the closest 15K sentences, selected using cosine similarity with source-side validation data.

To preprocess the data, we tokenize using the Moses tokenizer (Koehn et al., 2007) or the model-specific tokenizer provided by Huggingface. Additionally, we apply either Byte pair encoding (BPE) (Sennrich et al., 2015b) for the baseline transformer model and sentence piece (Kudo and Richardson, 2018) for all other models including mBART-50, M2M-100, NLLB-200, mT5-XL and mT5-XXL to split words into subwords tokens.

6 Experiments

This section describes the experimental details, including the toolkits, the parameter settings for the model training and decoding, and the results.

6.1 Tools & Hardware

For the Models mentioned in Section 4.2, we train the smaller models on 32GB NVIDIA Tesla V100 GPUs, and the medium and larger models require multiple 80GB NVIDIA A100 GPUs. We use a total of 4 V100 GPUs and 16 A100 GPUs. Due to GPU memory usage (see Section 1), we parallelized the training of the medium and larger models using the DeepSpeed package (Rasley et al., 2020).

6.2 Training Details

As an NMT baseline, we use the baseline transformer model (Vaswani et al., 2017) provided by the Fairseq toolkit. The model has half number of attention heads and the feed-forward network dimension compared to the Transformer (base) model in Vaswani et al. (2017). The rest of the network architecture is the same. We train this model from scratch by adding additional datasets and finally tuning it on the validation data.

We use the Fairseq toolkit for training the baseline transformer from scratch and for finetuning the mBART-50 and M2M-100 models. For finetuning NLLB-200, mT5-XL, and mT5-XXL models, we use the Huggingface toolkit. For the pre-trained multilingual models, we use the Hindi language encoder and English language decoder for finetuning and decoding.

As shown in Table 4, we finetune the models with the listed datasets for each subtask. We initially fine-tune these models on ID 4 dataset mentioned in Table 4. Finally, we further finetune the models on the validation datasets provided by the organizers.

Hyper-parameter settings We train the Transformer model from scratch and finetune all the multilingual pre-trained models. We train Transformer, mBART-50, and M2M-100 models for 10 epochs on the ID 4 datasets and 5 epochs on the validation dataset. We finetune the larger models listed in Table 3, for a maximum of 3 epochs before tuning on the validation for 7 epochs for subtask 1 and 4

Model	Train time/epoch	
	Subtask 1	Subtask 2
mBART-50	2 mins	14 mins
M2M-100	8 mins	33 mins
NLLB-200	16 mins	1.5 hrs
mT5-XL	20 mins	15 hrs
mT5-XXL	5.5 hour	24 hrs

Table 3: Per epoch training time for each of the models. The training time is for ID 4 datasets in Table 4.

epochs for subtask 2, respectively. We set the Adam betas to 0.9 and 0.98 for all the models and tuned the learning rates between $1e^{-5}$ and $9e^{-5}$. We opt for higher learning rates for the initial epochs and use lower learning rates for the remaining epochs. Finetuning with a high learning rate for fewer epochs is particularly helpful as larger models take much more time per epoch, even with the larger GPU memory. We also experiment with tuning the dropout between 0.1 and 0.15, and we get the best performance with the dropout rate set to 0.1. The batch size is limited to smaller values due to memory constraints. We set the batch size to 10 or 20 for larger models and 40 or 50 for medium-sized or smaller models.

Decoding parameters For the decoding step for both tasks, we set English as the target language tag for all the models. We tune the beam size, and the optimal beam size is 17 for both subtasks on the validation set. We limit the maximum sentence length to 128 only for the medium and larger models like NLLB-200, mT5-XL, and mT5-XXL. Finally, we detokenize the translation output as a post-processing step (Koehn et al., 2007).

6.3 Additional Experiments

We also perform additional experiments that are helpful but not included in the final submission due to limited time. These are the MTNT datasets and the ensemble methods. Firstly, we use the MTNT dataset as an additional bilingual in-domain data set containing different source languages. We also apply the multi-run ensemble method to combine models trained on multiple datasets together (Koehn and Knowles, 2017). For both tasks, we train M2M-100 models on the MTNT Fr-En data and the MTNT Ja-En data before tuning them on the baseline datasets, respectively. Additionally, we first fine-tune the WMT'14 News Hi-En data and then fine-tune the baseline data. Then we ensemble these two models with the original base model.

7 Results

We evaluate the models with respect to the BLEU score using `sacrebleu`. Table 5 shows the results of the experiments for both tasks and all the models. In general, we get improvement with larger multilingual models and with validation finetuning.

Table 4 shows the results of training from scratch using the transformer model with additional in-domain datasets. We get a maximum improvement of 9.3 for subtask 1 and 4.0 for subtask 2 using the additional datasets. Finally, tuning on validation gave an additional boost of +1.1 and +0.2 BLEU for subtasks 1 and 2 respectively. Table 5 shows the results for using pre-trained multilingual models on the ID 4 datasets. We get a maximum improvement of 25.6 and 32.6 for subtasks 1 and 2. This is +14.0 and +23.9 BLEU points higher than the best transformer model's results in Table 4.

Table 6 shows the ensemble results of a multi-run ensemble of the three models: (1) The baseline M2M-100 model in Table 5, (2) The M2M-100 model first trained on MTNT data and then on the baseline data, and (3) Training the M2M-model on MTNT data, then on WMT data, and finally on the baseline data. We get a slight decrease of −0.3 BLEU for subtask 1 compared to the baseline. However, for subtask 2, the performance improves by +0.8 BLEU points.

8 Analysis

We analyze the translation outputs of NLLB, mT5-XL, and mT5-XXL models. For subtask 1, the issues we faced were that the sentences were translated entirely to English and did not contain any Hinglish words. Some words were translated partially to Hinglish, and a portion of the words remained in the Hindi language. For subtask 2, the issues we faced were that the names of animal species were not translated correctly. And idioms lose their meaning in translation. Examples of these issues are shown in Table 7 & 8.

ID	Datasets	Hi-Hg
1	HinGE	1.2
2	[1]+Kaggle	6.4
3	[2]+WMT'14 News	10.3
4	[3]+Facebook MUSE	10.5
5	[4]+val tune	11.6

ID	Datasets	Hg-En
1	PHINC	4.5
2	[1]+HinGE	5.1
3	[2]+CALCS'21	5.2
4	[3]+Back-translation	8.5
5	[4]+val tune	8.7

Table 4: Adding in-domain datasets. Baseline: Transformer (Vaswani et al., 2017). Evaluation critierion: BLEU[%]. add citation of the datasets. Training from scratch without pre-trained models. '+val tune' is further finetuning on validation data. All the results are evaluated on the competition's test data.

Pretrained Multilingual Model	subtask 1		subtask 2	
	baseline	+val tune	baseline	+val tune
mBART-50	16.9	-	18.3	-
M2M-100	18.9	-	23.8	-
NLLB-200	11.5	-	23.8	30.3
mT5-XL	18.8	**25.6**	24.0	31.7
mT5-XXL	18.5	24.0	24.9	**32.6**

Table 5: Initialization with pre-trained models. BLEU scores (%) for subtask 1 and 2. 'baseline' experiment is finetuning the pre-trained model on the ID 4 datasets in Table 4. '+val tune' is further finetuning on validation data. All the results are evaluated on the competition's test data. **bold** results are the final submission.

Task	Models	BLEU
subtask 1	Base	18.9
	Base+MTNT+WMT	18.6
subtask 2	Base	23.8
	Base+MTNT+WMT	24.6

Table 6: Checkpoint ensemble results for subtask 2 trained on M2M-100 model evaluated on the competition's test data. The base is the baseline M2M-100 experiment. MTNT is first training on MTNT data and then tuning on the baseline. WMT tunes on MTNT, then WMT, and finally on baseline data.

Src	देश की राष्ट्रीय क्रिकेट टीम ...
NLLB	The national cricket team in the country...
mT5-XL	desh ki national cricket team...
mT5-XXL	country ki national cricket team...
Ref	desh ki national cricket team...
Src	यह प्रमाणित हो चुका है जो एक चमत्कार है ।
NLLB	It has been proven which is a miracle.
mT5-XL	yah pramanit ho chuka hai jo ek miracle hai.
mT5-XXL	yah pramanit ho chuka hai jo ek चमtkaar hai.
Ref	yah pramanit ho chuka hai jo miracle hai.

Table 7: Examples of errors for subtask 1.

9 Conclusion

This paper describes our submitted translation system for the WMT 2022 shared task MixMT competition. We train five different multilingual NMT models including mBART-50, M2M-100, NLLB-200, mT5-XL, and mT5-XXL, for both subtasks. We finetune on in-domain datasets including the validation data and significantly enhance our translation quality from 1.2 to 25.6 and 4.5 to 32.6 for subtasks 1 and 2 respectively. Additionally, we also apply data-augmentation techniques including back-translation, tuning on in-domain data, and checkpoint ensemble. Our system got the 1^{st} position in subtask 2 for both ROUGE-L and WER metrics, the 1^{st} position in subtask 1 for WER, and 3^{rd} position in subtask 1 for ROUGE-L.

Src	lol...gayi bhains paani mein...
NLLB	lol... went bhains in water...
mT5-XL	lol... animals went in water...
mT5-XXL	Lol... Goat got in the water...
Ref	lol.. buffalo went in the water...
Src	ye video dekh kar to khoon khaul gya
NLLB	After seeing this video, blood came out.
mT5-XL	seeing this video, my blood bleed.
mT5-XXL	Blood boiled after watching this video.
Ref	By watching this video, blood boiled.

Table 8: Examples of errors for subtask 2.

Acknowledgments

We appreciate the National Science Foundation (NSF) Award No. 1747728 and NSF CRAFT Award, Grant No. 22001 to fund this research. We are also thankful for the support of the Google Cloud Research Program. We especially thank Xuting Tang, Yu Yu, and Mengjiao Zhang to help editing the paper.

References

Gustavo Aguilar, Fahad AlGhamdi, Victor Soto, Thamar Solorio, Mona Diab, and Julia Hirschberg. 2018. Proceedings of the third workshop on computational approaches to linguistic code-switching. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. *arXiv preprint arXiv:1903.00089*.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Manish Shrivastava, and Dipti Misra Sharma. 2017. Joining hands: Exploiting monolingual treebanks for parsing of code-mixing data. *arXiv preprint arXiv:1703.10772*.

Ondřej Bojar, Vojtěch Diatka, Pavel Rychlỳ, Pavel Straňák, Vít Suchomel, Aleš Tamchyna, and Daniel Zeman. 2014. Hindencorp-hindi-english and hindi-only corpus for machine translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3550–3555.

Parth Chokhra. 2020. Hindi to hinglish corpus. https://www.kaggle.com/datasets/parthplc/hindi-to-hinglish.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Computing Surveys (CSUR)*, 53(5):1–38.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(107):1–48.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. *arXiv preprint arXiv:1601.01073*.

Saurabh Garg, Tanmay Parekh, and Preethi Jyothi. 2018. Code-switched language models using dual rnns and same-source pretraining. *arXiv preprint arXiv:1809.01962*.

Devansh Gautam, Prashant Kodali, Kshitij Gupta, Anmol Goel, Manish Shrivastava, and Ponnurangam Kumaraguru. 2021. Comet: Towards code-mixed translation using parallel monolingual sentences. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 47–55.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009.

Vikrant Goyal and Dipti Misra Sharma. 2019. Ltrc-mt simple & effective hindi-english neural machine translation systems at wat 2019. In *Proceedings of the 6th Workshop on Asian Translation*, pages 137–140.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor OK Li. 2018. Universal neural machine translation for extremely low resource languages. *arXiv preprint arXiv:1802.05368*.

Ganesh Jawahar, El Moatez Billah Nagoudi, Muhammad Abdul-Mageed, and Laks VS Lakshmanan. 2021. Exploring text-to-text transformers for english to hinglish machine translation with synthetic code-mixing. *arXiv preprint arXiv:2105.08807*.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Philipp Koehn. 2020. *Neural machine translation*. Cambridge University Press.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In

Proceedings of the First Workshop on Neural Machine Translation, pages 28–39. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Sahinur Rahman Laskar, Abinash Dutta, Partha Pakray, and Sivaji Bandyopadhyay. 2019. Neural machine translation: English to hindi. In *2019 IEEE conference on information and communication technology*, pages 1–6. IEEE.

M. Paul Lewis, editor. 2009. *Ethnologue: Languages of the World*, Sixteenth edition. SIL International, Dallas, Texas, USA.

Paul Michel and Graham Neubig. 2018. Mtnt: A testbed for machine translation of noisy text. *arXiv preprint arXiv:1809.00388*.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2018. Improving neural machine translation by incorporating hierarchical subword features. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 618–629.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553.

Surangika Ranathunga, En-Shiun Annie Lee, Marjana Prifti Skenduli, Ravi Shekhar, Mehreen Alam, and Rishemjit Kaur. 2021. Neural machine translation for low-resource languages: A survey. *arXiv preprint arXiv:2106.15115*.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Thamar Solorio, Shuguang Chen, Alan W Black, Mona Diab, Sunayana Sitaram, Victor Soto, Emre Yilmaz, and Anirudh Srinivasan. 2021. Proceedings of the fifth workshop on computational approaches to linguistic code-switching. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*.

Vivek Srivastava and Mayank Singh. 2020. PHINC: A parallel Hinglish social media code-mixed corpus for machine translation. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 41–49, Online. Association for Computational Linguistics.

Vivek Srivastava and Mayank Singh. 2021. HinGE: A dataset for generation and evaluation of code-mixed Hinglish text. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 200–208, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xu Tan, Jiale Chen, Di He, Yingce Xia, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with language clustering. *arXiv preprint arXiv:1908.09324*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Tatoeba. 2022. Spanish english bilingual dataset. `https://www.manythings.org/anki/`.

Louis Tom. 2022. Codemixed. `https://www.kaggle.com/datasets/louistom/codemixed`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Raúl Vázquez, Alessandro Raganato, Jörg Tiedemann, and Mathias Creutz. 2018. Multilingual nmt with a language-independent attention bridge. *arXiv preprint arXiv:1811.00498*.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2019. Code-switched language models using neural based synthetic data from parallel sentences. *arXiv preprint arXiv:1909.08582*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Hao Xiong, Junchi Yan, and Li Pan. 2021. Contrastive multi-view multiplex network embedding with applications to robust network alignment. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1913–1923.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018. A dataset for document grounded conversations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 708–713, Brussels, Belgium. Association for Computational Linguistics.

The University of Edinburgh's Submission to the WMT22 Code-Mixing Shared Task (MixMT)

Faheem Kirefu **Vivek Iyer** **Pinzhen Chen** **Laurie Burchell**
School of Informatics, University of Edinburgh
{fkirefu,vivek.iyer,pinzhen.chen,laurie.burchell}@ed.ac.uk

Abstract

The University of Edinburgh participated in the WMT22 shared task on code-mixed translation. This consists of two subtasks: i) generating code-mixed Hindi/English (Hinglish) text generation from parallel Hindi and English sentences and ii) machine translation from Hinglish to English. As both subtasks are considered low-resource, we focused our efforts on careful data generation and curation, especially the use of backtranslation from monolingual resources. For subtask 1 we explored the effects of constrained decoding on English and transliterated subwords in order to produce Hinglish. For subtask 2, we investigated different pretraining techniques, namely comparing simple initialisation from existing machine translation models and aligned augmentation. For both subtasks, we found that our baseline systems worked best. Our systems for both subtasks were one of the overall top-performing submissions.

1 Introduction

Code-mixing is the shift from one language to another within a single conversation or utterance (Sitaram et al., 2019). It is an extremely common and diverse communicative phenomenon worldwide (Doğruöz et al., 2021; Sitaram et al., 2019), though one which is currently under-served by many NLP technologies (Solorio et al., 2021).

One of the most well-known examples of code-mixing is between Hindi and English, commonly referred to as Hinglish[1]. It is extremely common amongst Hindi-English bilingual speakers in both speech and text, used across a range of genres and media (Parshad et al., 2016), and has its own distinctive features and linguistic forms (Kumar, 1986; Sailaja, 2011). The process of generating Hinglish from the written text is non-trivial, as code-mixing may happen at the phrase or word level, but Hindi and English differ substantially syntactically.

As a novel addition to the current code-mixing NLP research, we investigated lexically constraining the Hinglish output in subtask 1 to only contain words from English and Hindi sources. Through analysis, we demonstrated that transliteration mismatches could affect performance.

Another novel approach we explore for this task, particularly for subtask 2, is a denoising-based pretraining technique called Aligned Augmentation (AA) (Pan et al., 2021). AA, which trains MT models to denoise artificially generated code-mixed text, was shown by Pan et al. (2021) to boost translation performance across a variety of languages - thanks to the enhanced transfer learning brought about by code-mixed pretraining. In this work, we explored if this general-purpose approach could be useful for translating authentic, human-generated code-mixed text, focusing on Hinglish.

Despite these efforts, we found that for both subtasks our original baselines worked better and constituted our final submissions for this task, which ranked as one of the top-performing systems for both subtasks, by both automatic and human evaluation. We hope our methods, particularly Hinglish data generation, that allowed us to build these systems would be useful to the community; as would the findings from our additional research explorations.

2 Related Work

2.1 Code-mixing

Due to an increasing prevalence of code-mixed data on the Internet, there is a growing body of research into code-mixing, particularly for Hinglish, in the NLP community. Doğruöz et al. (2021) provide a comprehensive literature review of code-mixing in the context of language technologies. Whilst they highlight several challenges inherent

[1] In the scope of this paper, we designate "hg" as the language code for Hinglish.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1145 - 1157
December 7-8, 2022 ©2022 Association for Computational Linguistics

in NLP with code-mixed text (such as understanding cultural and linguistic context, evaluation, and a lack of user-facing applications), the most notable obstacle for this shared task is the lack of data. They note that there are very few code-mixed datasets, making it challenging to build deep learning models such as those for NMT. In this work, we use backtranslation as our main data augmentation method (Edunov et al., 2020; Barrault et al., 2020; Akhbardeh et al., 2021, *inter alia*). This allows us to leverage the larger amount of monolingual data for better final model performance. The XLM toolkit (Lample and Conneau, 2019) seemed an ideal choice to backtranslate our Hinglish. This is because it has shown promising results in unsupervised and semi-supervised settings where parallel data is sparse, but monolingual data is ample. Also given that Hinglish is closely related to both languages, we believed Hinglish should be an ideal language to use in a semi-supervised setting.

2.2 Constrained decoding

Constrained decoding involves applying restrictions to the generation of output tokens during inference. Most implementations have the goal of ensuring that desired vocabulary items appear in the target side sequence (Hokamp and Liu, 2017; Hasler et al., 2018; Post and Vilar, 2018). Alternatively, Kajiwara (2019) paraphrase an input sentence by forcing the output to not include source words, and Chen et al. (2020) constrain NMT decoding to follow a corpus built in a trie data structure to find parallel sentences.

To the best of our knowledge, previous linguistics research investigated and applied the grammatical constraints in code-mixing (Sciullo et al., 1986; Belazi et al., 1994; Li and Fung, 2013), rather than the novel method in our work of introducing lexical constraints.

2.3 Aligned augmentation

Several recent works (Yang et al., 2020a,b; Lin et al., 2020; Pan et al., 2021) have explored enhancing cross-lingual transfer learning by pretraining models on the task of 'denoising' artificially code-mixed text. Methods to create the necessary code-mixed data vary, and include bilingual or multilingual datasets and word aligners (Yang et al., 2020a, 2021), lexicons (Yang et al., 2020b; Lin et al., 2020; Pan et al., 2021), or combining code-mixed noising with traditional masked noising approaches (Li et al., 2022).

The most successful among these methods is Aligned Augmentation (AA) (Pan et al., 2021), which randomly substituting words in the source sentence with their word-level translations, as obtained from a MUSE (Lample et al., 2018) dictionary. Pan et al. (2021) showed that their technique can effectively align multilingual semantic word representations and boost performance across various languages. However, these methods focus on training general-purpose MT models. In this work, we investigate their utility for translating real human-generated code-mixed text.

2.4 Automatic evaluation metrics

Automatic translation evaluation is usually done using BLEU (Papineni et al., 2002), yet there is no comprehensive study on its suitability for code-switched translation. Specifically in this task, the organisers announced that the participating systems will be evaluated using ROUGE-L (Lin, 2004) and word error rate (WER). Nonetheless, the packages implementing these metrics were not specified. Since ROUGE comes with different language, stemming and tokenisation settings, we instead used BLEU, ChrF++ (Popović, 2017), translation error rate (TER), and WER[2] for our internal validation. The first three are as implemented with sacreBLEU (Post, 2018). We stick to the default configurations, except that the ChrF word n-gram order is explicitly set to 2 to make it ChrF++. In addition, the organisers performed a small-scale human evaluation on 20 test instances for all submissions.

In this work, we advocate for a character-based metric when evaluating the Hinglish output in subtask 1. This is because for the code-switched language, there is no formal spelling or defined grammar, and words may have a diverse range of acceptable transliterations and lexical forms.

3 Subtask 1: Translating into Hinglish

Good quality Hinglish data is hard to come by, and parallel Hinglish data with Hindi or English even more frugal. Therefore, for both subtasks we concentrated our efforts on generating good Hinglish backtranslation. We planned to use the model which produced the highest quality Hinglish for subtask 1 as our backtranslator for subtask 2, hence we focused our efforts on each subtask sequentially.

[2] https://github.com/jitsi/jiwer

3.1 Data cleaning and preprocessing

After deduplicating the data, we removed non-printing characters and normalised the punctuation. We then ran rule-based filters, removing any sentences with fewer than two or more than 150 words, where fewer than 40% of the words are written in the relevant script, or where over 50% of characters are not letters in the relevant script. For English and Hindi, we ran `fasttext` language ID and removed any sentence which was not classified as the relevant language.[3] For Hinglish, we also removed any sentence with a predicted probability of English greater than 0.99 in order to remove sentences that were solely in English. We tokenised English and Hinglish using Moses scripts (Koehn et al., 2007) and tokenised Hindi using the `indicnlp` library (Kunchukuttan, 2020).

We decided to add explicit preprocessing and postprocessing capabilities for handling social media text, given that this was the domain for subtask 2. On both source and target sides, we replaced URLs, Twitter handles, hashtags and emoticons each with their own placeholder tokens[4], to be replaced back from the source after inference. These placeholders made up 1.7% of the validation set tokens for subtask 2, far higher than would appear in general domain data.

3.1.1 The HinGe dataset

The primary dataset for subtask 1 was the HinGe dataset (Srivastava and Singh, 2021), which consisted of hi-en-hg parallel sentences, with some examples synthetic and some human-generated. This was provided to us pre-split into training and development sets for both data types. However, we noticed that these sets were not mutually exclusive, and after deduplication and filtering on the synthetic data human annotations[5], we had 6,727 hi-en-hg examples in total.

3.1.2 Base hi↔en translation models

Firstly, we trained four Transformer-base (Vaswani et al., 2017) models with different seeds using Marian (Junczys-Dowmunt et al., 2018) for both hi→en and en→hi directions, using the data from the hi-en parallel Samanantar corpus[6] (Ramesh et al., 2021). Given the findings of Ding et al. (2019) with regard to vocabulary choice for low-resource scenarios, and that our task inherently contains transliteration, we opted for a low BPE (Sennrich et al., 2016) merge size of 4k, resulting in a small joint vocabulary of 7.9k. We used the hi-en FLORES development set (Goyal et al., 2022) for validation and early stopping, and noticed our model produced surprisingly good quality translations in both directions[7]. We used these models (along with vocabulary) to both initialise subsequent models and generate backtranslation for more training data.

3.1.3 Hinglish data

L3Cube-HingCorpus (Nayak and Joshi, 2022) and CC-100 Hindi Romanized (Conneau et al., 2020a) are two Hinglish corpora that we wished to backtranslate into both English and Hindi. Given that we only had a small amount of parallel Hinglish data, compared to our 'monolingual' datasets, we used the XLM toolkit (Lample and Conneau, 2019) to train a semi-supervised model (see Appendix A for details). We then backtranslated the monolingual Hinglish data into both English and Hindi. However, given the noisy quality of the data and translations themselves, we decided to evaluate them using our hi→en and en→hi Marian models. Specifically, for an en-hi backtranslated (XLM) sentence pair, we translated the en/hi into hi/en respectively, then evaluated the double translated output using ChrF, with the XLM backtranslations as the references. We then took a mean of the English and Hindi ChrF score to get our final confidence value. We used the resulting hg-en-hi sentence trios with values at least 0.4, to compromise between the quality and quantity of data available to use as training. Most of the sentences scored quite poorly, and filtering on 0.4 yielded 2.1M sentences, only about 12% of the original Hinglish monolingual dataset.

3.1.4 Transliteration

In order to best leverage the Samanantar hi-en parallel corpus, we transliterated the Hindi side into Roman script[8], on the word level. Although this

[3] Our cleaning scripts are adapted from those provided by the Bergamot project. https://github.com/browsermt/students/tree/master/train-student Specifically, we add support for Hindi and Hinglish text.

[4] `<URL>, <TH>, <HT>` and `<EMO>` respectively

[5] We only kept sentences with an average rating greater than 4, and annotator disagreement less than 5

[6] Each sentence was annotated with the LaBSE (Feng et al., 2022) Alignment Score (between 0 and 1), so we filtered out values less than 0.65, resulting in around 10.1M sentences

[7] sacreBLEU: 33.8 for hi→en and 32.7 for hi→en on FLORES development set

[8] In the scope of this paper, we use "ht" to denote pure romanised Hindi transliteration

Beam Size	BLEU (↑)	ChrF++ (↑)	TER (↓)	WER (↓)
Unconstrained				
1	17.8	42.8	65.3	**81.5**
4	**18.1**	**44.0**	**64.5**	85.7
12	18.0	43.8	64.8	86.0
24	18.0	43.7	65.0	85.5
36	17.9	43.5	65.1	85.4
48	18.0	43.6	65.2	85.5
Constrained				
1	10.8	33.1	76.1	75.1
2	12.2	35.6	74.9	69.1
4	13.2	36.6	74.2	63.5
6	14.1	37.7	73.5	60.8
12	14.6	38.1	73.7	58.6
24	14.8	38.5	73.5	57.2
36	14.9	38.7	73.6	**56.7**
48	15.0	38.7	73.6	57.0

Table 1: Experimental results on the validation set with unconstrained and constrained decoding for subtask 1.

forward transliteration was not likely to contain much code-mixed text, it would still be useful training data for our model, given that both the Hindi and English sources are assumed to be either the original sources or human translationese.

We used the AI4Bharat Indic transliterator (Madhani et al., 2022), to convert (on the word level) all romanised tokens contained in our monolingual Hinglish datasets into Devanagari script. This tool is a neural-based model with beam search capabilities, therefore we generated the top 4 results in Hindi for each Hinglish token. We used the top 4 instead of the most likely candidate as, upon inspection, we found that the correct corresponding Hindi token was not always predicted first. We also used a human-generated list of Hinglish-English pairs form the Xlit-Crowd corpus (Khapra et al., 2014) which we treated as the gold standard.

To summarise, our training data for our hi→ht transliterator[9] consists of 5.3M Hinglish-Hindi word pairs (1.3M unique Hinglish words), and 15k from XlitCrowd, of which we use 1k as a validation set for early stopping. We train a small transformer model with Marian on the **character-level** for both input and output. When forward transliterating the Hindi side of the Samanantar corpus, we copied over non-standard strings (such as numbers, punctuation etc.), or else we looked up the token (if it existed) in our gold standard list. Otherwise, we used our transliteration model as a final back-off. In hindsight, one disadvantage of our approach was that we did not generate multiple candidates for each Hindi word, to reflect the diversity of possible romanised candidate tokens.

We also used this transliteration model as part of our constrained decoding experiments later (see Section 3.3).

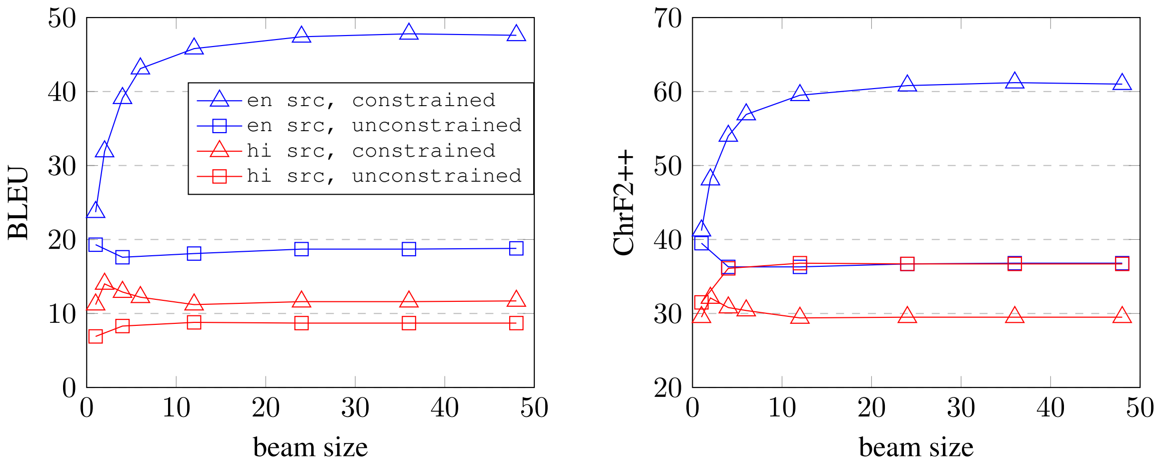
3.2 Baseline (unconstrained decoding)

We decided to use a dual encoder setting given that we have two inputs in this task, and initialise our model from our previously trained Marian MT systems. We used hi→en to initialise the Hindi-decoder and the English-encoder cross attention parameters, whereas en→hi was used to initialise the English-encoder and all other decoder parameters. Our vocabulary was the same as the pretrained models.

Early stopping with patience 10 on the HinGe dataset was used for convergence - for all of the experiments mentioned in this paper. Our training regime consisted of two stages:

- General domain - The training datasets used were the backtranslated Hinglish and forward transliterated Samanantar corpora. We used all of the HinGe dataset as a validation set.
- Finetuning - We continue training on a sub-

[9] We decided to build our own transliterator as we found existing tools in this direction to be of poor quality



(a) BLEU

(b) ChrF++

Figure 1: Validation BLEU and ChrF++ of the constrained and unconstrained outputs scored against English and transliterated Hindi *sources* separately.

set of HinGe dataset, using a distinct smaller subset (1k) of it as a validation set.

3.3 Constrained decoding

After analysing the training data, we hypothesized that nearly all the output words should either be from the English source, or as a transliteration of a word from the Hindi source, with likely little change in sentence structure. This inspired us to use the technique of constrained decoding when generating Hinglish.

Unlike standard constrained decoding where a model is forced to incorporate certain words in the output, our proposal is to exclude vocabulary words that do not exist in English or transliterated Hindi source sentences. Following Chen et al. (2020)'s notion, we applied pre-expansion pruning: disallowed word paths are assigned an extremely small score before hypotheses are ranked and expanded. Specifically, to obtain Hindi transliteration, we used our transliteration model described in Section 3.1.4.

We performed beam searches with constrained decoding and reported automatic scores on the validation set in Table 1. Unfortunately, constrained decoding does not beat unconstrained decoding. As a general trend, WER and TER do not change much as beam size increases, while BLEU and ChrF++ significantly improve.

To better understand the impact of constrained decoding, we score the validation outputs against English and transliterated Hindi sources separately, then plot BLEU and ChrF++ numbers in Figure 1a and Figure 1b. We observe that with increasing beam sizes, constrained decoding prefers to generate English tokens instead of transliterated Hindi. Unconstrained decoding achieves a much better balance.

One hypothesis is that the quality of Hindi transliteration is not perfect, resulting in the model preferring English tokens from the vocabulary. Hence, we compute the percentage of words in the gold reference as well as in the unconstrained (baseline) output that come from neither the English nor the transliterated Hindi source. Surprisingly, on average 45.1% of the total words in the unconstrained output do not appear in the sources; as for the gold reference, it is 39.8% which is slightly lower. It is worth noting that the numbers might be inflated as we computed the word overlap after outputs are detokenised. Yet it implies that many of the reference words do not exactly appear in the lexical constraints determined from the source senteneces.

Finally, we visualise the first five validation sentences in Table 2. We highlight in *red* the target words that do not exist in the source sentences; we also label the possible corresponding tokens from the sources in *blue*. It can be confirmed that most mismatches are due to differences in Hindi transliteration and letter cases. This indicates that the lexical constraint idea is suitable in theory, but it is hindered by the error propagation in transliteration. This may have been alleviated by running multiple transliteration schemes on the Hindi source to make the constraints more diversified.

hi source	1995 से 2004 के दौरान औसत धरातलीय तापमान 1940 से 1980 तक के औसत तापमान से भिन्न्न है
hi transliteration	1995 *sey* 2004 ke *Dauran* ausat *Dharatliya Tapman* 1940 *sey* 1980 tak ke ausat *Tapman sey bhinnn is*
en source	The average *geological* temperature of the earth from 1995-2004 is different than that of 1940-1980 .
constrained	The average Dharatliya tapman of the earth from 1995 -tak ke ausat tapman from bhinn.
unconstrained	1995 *se pratik dauran* average *dharatliy temperwof* the earth from *1990 se* 1980 tak ke ausat *temperwale se bhinn hai*.
reference	from 1995-2004 ke *dauran* average *geology* temperature of earth 1940 *se* 1980 tak ke ausat temperature *se* different *hai*.
hi source	धृतराष्ट्र एवं गांधारी के १०० पुत्रों में सबसे बड़े ।
hi transliteration	Dhritrashtra Evan Gandhari ke 100 putron *main* sabse bade .
en source	Dhrudharashtra and Ghandhari 's eldest among their 200 sons .
constrained	Dhrudharashtra among their 200 sons.
unconstrained	Dhrudharashtra among their 200 sons.
reference	Dhrudharashtra and Ghandhari ke 100 sons *mein* sabse bade.
hi source	इस प्रकार राजस्थान के रेगिस्तान का एक बड़ा भाग शस्य श्यामला भूमि में बदल जायेगा ।
hi transliteration	is *Prakar* rajasthan ke registan ka a badaa bhaag Shasya Shyamala bhumi *main* cange jayega .
en source	In this way a major part of the desert in Rajasthan would become a harvesting and fertile land .
constrained	In this way a major part of the desert in Rajasthan would become a harvesting and jayega.
unconstrained	In this way a major part of the desert in Rajasthan would become a harvesting and *wtile* land.
reference	is *prakar* rajasthan ke desert ka *ek* major part harvesting and fertile land *mein badal* jayega.
hi source	राष्ट्रपति की अध्यादेश जारी करने की शक्ति पे नियंत्रण
hi transliteration	Rashtrapati ki *Adhyadesh jaari* karne ki shakti pay *Niyantran*
en source	The power of the President to proclaim *Ordinance* is subject to :
constrained	Rashtrapati ki Adhyadesh jaari karne ki
unconstrained	Rashtrapati ki *adhyadesh* jaari karne ki *pratiniyantran*.
reference	President ki *ordinance jari* karne ki power *pr niyantran*.
hi source	1000 से अधिक हाथी निर्माण के दौरान यातायात हेतु प्रयोग हुए थे ।
hi transliteration	1000 *sey Adhik* haathi *Nirman* ke *Dauran* yatayat hetu *pryog huye* they .
en source	*More* than 1000 elephants were used during the time of construction for transportation .
constrained	Dauran transportation ke time yatayat hetu pryog hue the.
unconstrained	1000 *se adhik* haathi *nirman* ke *dauran* transportation hetu pryog *hue* the.
reference	*more* than 1000 elephants construction ke *dauran* transportation hetu *prayog hue* the.

Table 2: The first five validation instances: English and Hindi sources, as well as constrained, unconstrained and reference outputs. *red* denotes the target side words that do not appear in either of the source sentences from a constrained aspect; *blue* denotes possible source-target matches in a different surface form.

4 Subtask 2: Hinglish-to-English

4.1 Data cleaning and preprocessing

The primary dataset provided for this task PHINC (Srivastava and Singh, 2020) is relatively small at 13.7k English-Hinglish pairs. Therefore, we aimed to generate domain-specific parallel data using our baseline model from Subtask 1 on English monolingual data.

We analysed the source side of the validation dataset to determine the most frequent content words (see Table 3) and then selected these words (and any morphological/spelling variants) from the English WikiMatrix corpus (Schwenk et al., 2021). This yielded a total 477k English sentences and we henceforth refer to this selection of sentences as ToxicWiki. We also used Sentiment140 (Sahni et al., 2017), a dataset of 1.6M tweets in English, as the domain of our validation set is also Twitter.

Word	Validation	WikiMatrix
rape	249	23,198
hate	117	16,824
terrorism	24	11,160
khoon (blood)	21	59,526
murder	21	75,066
india	16	291,054
Total	-	476,828

Table 3: Frequency of top content words present in our validation set, and the number of sentences within WikiMatrix that contained the word (or morphological variants). The resulting sentences formed ToxicWiki

To obtain the Hinglish side of both Sentiment140 and ToxicWiki datasets, we backtranslated into Hindi using our en→hi Marian model, and then used the en-hi pair and our baseline system for subtask 1 to obtain the corresponding Hinglish. However, many of the placeholders (such as <HT>) did not occur frequently enough during the training of subtask 1 for the model to learn to consistently copy them across; therefore the model was not able to predict them with a large degree of accuracy. Therefore, we ran a postprocessing script that corrected for placeholders on the backtranslated Hinglish, given the English source, so that our downstream model would be able to learn to simply copy these placeholders across. Specifically, we made sure that the number of each placeholder type in the backtranslated Hinglish was the same (and in roughly the same position) as that in the source sentence.

For the AA experiments described in Section 4.3, we used monolingual Hindi, English and Hinglish data. For Hindi and English, we randomly sampled 20M sentences from the News Crawl corpora (Akhbardeh et al., 2021). For Hinglish, the monolingual corpora described above was used. In order to code-mix these corpora as described in the AA algorithm, we used MUSE dictionaries for the Hindi-English pair. For Hinglish-Hindi pairs, we used the data generated with AA for the transliteration model.

4.2 Baseline systems

We used a hi→en MT to initialise the baseline hg→en model.

Our training regime consisted of three stages:

1. General - Training on the backtranslated en-hg internet corpora (with confidence value at least 0.4), and ht-en side of the Samanantar corpus, where we treat the transliteration as Hinglish. We used the PHINC dataset as our validation set for early stopping.

2. We continued training on Sentiment140 and ToxicWiki corpus, using the same validation set as before, until convergence.

3. We continued training on the PHINC dataset, using a small subset (1k) of it as validation data for early stopping.

As we had multiple hi→en MT systems, we also trained an ensemble model of four, where we followed the same training regime above with parameters initialised from each of our hi→en models. Our results are shown in Table 4, with our ensemble model outperforming the single on all metrics.

4.3 Aligned Augmentation for subtask 2

Our Aligned Augmentation (AA) experiments where implemented with Fairseq (Ott et al., 2019), and we used the Transformer architecture, with 12 encoder and 12 decoder layers. Our first step consisted of pretraining these models on Hindi, English, and Hinglish corpora, with the target being the "denoised" sentence - thus training the model to reconstruct the original sentence, following the AA algorithm. For validation, we randomly sampled 1k sentences from the training corpus.

We then finetuned this model on the Hinglish-English parallel corpora mentioned above. The major AA baselines we trained and their performances are listed in Table 5 - along with a randomly initialised baseline that was trained solely on the parallel corpora. The data sources we used in our experiments were quite diverse: we started with high-quality monolingual data for pretraining followed by parallel datasets of varying domains and qualities, (the Hinglish backtranslated corpora, Sentiment140, PHINC and ToxicWiki). We attempted to explore how best these resources could be utilised. We started with our default training paradigm: we finetuned on backtranslated Hinglish, followed by the ToxicWiki and then a shuffled concatenation of the social media datasets - the Sentiment140 and PHINC datasets respectively. This was based on the intuition that the final model should be most recently trained on datasets from similar domains as the test set.

Following this paradigm, we conducted two sets of experiments: a "validation experiment" that tries to estimate the best choice of validation sets, and "training experiments" to verify the importance of some training sources empirically. The former is a crucial decision in our experiments given our use of early stopping. We find that validating on the official MixMT validation sets released for Subtask 2 ends up performing significantly worse than validating on a subset of the respective training datasets. This is surprising given the performances reported in Table 5 are evaluated on the same validation sets. This suggests that training and validating the model on corpora from different domains can help boost the final performance - even if it does not improve loss on the final validation set. In the latter body of experiments, we attempted to determine the value of the XLM backtranslated corpora on performance - which seems very noisy on manual inspection, with the target side (English) being

	BLEU (↑)	ChrF++ (↑)	TER (↓)	WER (↓)
Baseline Experiments				
Single model	24.5	47.0	65.1	72.0
Ensemble (of 4)	**25.5**	**48.7**	**62.9**	**70.5**

Table 4: Baseline results for subtask 2 on the MixMT validation set.

generated through backtranslation. Surprisingly, its inclusion significantly enhances performance, by +5 BLEU points. This could be due to various reasons: its sheer size (15M sentences), the presence of word-level translations between English to Hinglish in parallel sentences (despite grammatical errors), the similarity between the source and the target encouraging “copying" which can sometimes be beneficial for this task, etc. We also find that the inclusion of hi-en along with hg-en further boosts performance, consistent with the findings of previous works on multilingual MT. We empirically found that including ‘all’ available hi-en sentences and ‘all’ available hg-en sentences was more beneficial than splitting our parallel dataset into the two respective directions – despite the target sentence being duplicated in the former.

Compared to the Random baselines, our final AA baselines show consistent improvement for all given metrics - though the improvement is not very significant with respect to BLEU o TER. A closer glance at the validation set and the generated predictions reveals the potential reason behind this - there is a significant amount of noise present in the validation sets due to the social media domain, with errors in both syntax and semantics. Given that it is not always easy to comprehend and translate such sentences well, the gold reference sentences are sometimes of relatively poor quality - containing various potential errors such as inaccurate word form predictions, grammatical errors, misspellings etc. While word-based metrics may fail to handle these cases; ChrF++, being a character-based metric, can likely alleviate noise that may have propagated to reference sentences and might be a more suitable metric for Subtask 2 as well. It is encouraging to note AA’s improvement over the Random baseline in this light.

AA appears to bring about some improvement qualitatively, especially regarding noisy input - for instance, it was able to more accurately translate misspellings and handle grammatical inconsistencies. However, the frequency of sentences where AA performs better than its randomly initialized counterparts seems relatively low. One explanation could be that fine-tuning the model on 18M parallel sentences could lead it to ‘forget’ the representations learned during pretraining. This is in line with the findings of (Pan et al., 2021) that observe relatively lower improvements for high-resource languages. While adding large corpora (15M sentences) such as the XLM backtranslated corpora does lead to net improvements, it is possible optimization in the size of finetuning data used could lead to even greater gains. Secondly, given that our dictionaries appear to help in noise resolution, it might be useful to incorporate various types of misspellings rigorously in the code-mixing lexicons created - thus enabling the final model to be more robust. Finally, including training corpora from other Indo-Aryan languages like Urdu or Marathi could be beneficial. Although Subtask 2 focuses on the translation of Hinglish-English, the validation and test sets (as well as training sets) contain many examples of code-mixing between related Indo-Aryan languages and English - most prominently in Urdu, which is historically and linguistically similar to Hindi.

In the end, we observe that the AA models we train are unable to beat our original single-model baseline, despite having more parameters. Curiously, this is also the case for the randomly initialized baseline in Table 5. Due to time constraints, we are unable to investigate the reasons behind these. Possible explanations could include: training paradigm differences (initializing with hi→en vs mixing hi→en with hg→en), ensembling, experimental setting disparities, inherent differences between training libraries (Fairseq vs Marian). It is possible that addressing these disparities, as well as exploring the directions suggested in the previous paragraph, could enable AA baselines to yield superior results for code-mixed translation.

	BLEU (↑)	ChrF++ (↑)	TER (↓)	WER (↓)
Validation Experiments				
AA (dev = MixMT valid)	20.5	41.2	72.7	78.6
AA (dev = train subset)	23.3	45.7	68.3	74.6
Training Experiments (dev=train subset)				
AA (train = all Hg->En minus XLM BT data)	18.3	38.4	78.3	83.4
AA (train = all Hg->En)	23.3	45.7	68.3	**74.6**
AA (train = all Hg->En + all Hi->En)	**24.4**	**46.2**	**68.2**	74.9
Random	24.3	45.2	68.4	74.6

Table 5: Aligned Augmentation experiments for subtask 2, as evaluated on the official MixMT Subtask 2 validation set. “Validation experiments" refers to experiments performed to select the best choice of the validation set for early stopping. ‘MixMT valid’ refers to the same validation set mentioned earlier (that is also used for evaluation), while ‘train subset’ refers to a subset (last 1000 sentences) of the respective training corpus. “Training experiments” seek to explore various dataset choices during training time, using a subset from the training corpus for validation.

	BLEU	ChrF++	TER	WER	ROUGE-L	Human Eval. Score
Subtask 1	26.9	52.7	55.2	56.2	57.9	3.85
Subtask 2	28.7	51.2	59.1	61.3	62.5	3.75

Table 6: Final Test Results for the University of Edinburgh’s submissions of MixMT 2022. BLEU, ChrF++ and TER were evaluated by us while WER and ROUGE-L results are from the official Codalab leaderboard. Human evaluation (on a scale of 1-5) was provided by the organisers on 20 random sentences and we report the average.

5 Test Results

The final test results for our submissions are listed in Table 6. For Subtask 1, we used unconstrained decoding with beam-size 12, and for Subtask 2 we used our baseline ensemble (4) with beam-size 36. We evaluated BLEU, ChrF++ and TER ourselves, while the other metrics are provided by the organizers. We ranked second in both subtasks on the MixMT leaderboard[10] although in both the automatic and human evaluation[11], there does not appear to be a statistically significant difference. Furthermore, we note that some participants have an exceedingly high number of test submissions and would encourage future shared tasks to put in place measures to avoid this.

6 Conclusion

In this work, we described our various findings and experiences while building NMT systems that translated between Hinglish and monolingual English/Hindi - as part of the WMT22 Code-Mixing Shared Task. We proposed various corpora that could be useful for these tasks - many of which we create as part of this work - and utilizing these, build high-performing MT systems that, for both subtasks, constituted one of the leading unconstrained models. In addition, we also explored and analysed some alternative approaches for training our models like constrained decoding and Aligned Augmentation (AA) which, despite not beating our original baselines, yielded findings that are useful for future research. Perhaps the most notable of these suggests that efforts to create Hinglish datasets, including using transliterated Hindi as an approximation, can be fruitful and pivotal to high performance. While efforts to handle noise in social media text (such as AA-based pretraining) can also help, further research is required to establish the most optimal ways to do the same.

7 Acknowledgements

We would like to give special thanks to Nikita Moghe for her valuable feedback and insights throughout this research.

This work was supported in part by the UKRI Centre for Doctoral Training in Natural Language Processing, funded by the UKRI (grant EP/S022481/1) and the University of Edinburgh, School of Informatics. The experiments in this pa-

[10] https://tinyurl.com/codalab-ldbd

[11] https://tinyurl.com/heval-mixmt

per were performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service[12], and using the Sulis Tier 2 HPC platform hosted by the Scientific Computing Research Technology Platform at the University of Warwick. Sulis is funded by EPSRC Grant EP/T022108/1 and the HPC Midlands+ consortium.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Hedi M. Belazi, Edward J. Rubin, and Almeida Jacqueline Toribio. 1994. Code switching and x-bar theory: The functional head constraint. *Linguistic Inquiry*, 25(2):221–237.

Ondřej Bojar, Vojtěch Diatka, Pavel Rychlý, Pavel Straňák, Vít Suchomel, Aleš Tamchyna, and Daniel Zeman. 2014. HindEnCorp - Hindi-English and Hindi-only Corpus for Machine Translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Pinzhen Chen, Nikolay Bogoychev, Kenneth Heafield, and Faheem Kirefu. 2020. Parallel sentence mining by constrained decoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1672–1678, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Shuoyang Ding, Adithya Renduchintala, and Kevin Duh. 2019. A call for prudent choice of subword merge operations in neural machine translation. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 204–213, Dublin, Ireland. European Association for Machine Translation.

A. Seza Doğruöz, Sunayana Sitaram, Barbara E. Bullock, and Almeida Jacqueline Toribio. 2021. A survey of code-switching: Linguistic and social perspectives for language technologies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1654–1666, Online. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Marc'Aurelio Ranzato, and Michael Auli. 2020. On the evaluation of machine translation systems trained with back-translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2836–2846, Online. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Trans. Assoc. Comput. Linguistics*, 10:522–538.

[12] www.csd3.cam.ac.uk

Eva Hasler, Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 506–512, New Orleans, Louisiana. Association for Computational Linguistics.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Tomoyuki Kajiwara. 2019. Negative lexically constrained decoding for paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6047–6052, Florence, Italy. Association for Computational Linguistics.

Mitesh M. Khapra, Ananthakrishnan Ramanathan, Anoop Kunchukuttan, Karthik Visweswariah, and Pushpak Bhattacharyya. 2014. When transliteration met crowdsourcing : An empirical study of transliteration via crowdsourcing using efficient, non-redundant and fair quality control. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 196–202. European Language Resources Association (ELRA).

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

Ashok Kumar. 1986. Certain aspects of the form and functions of Hindi-English Code-Switching. *Anthropological Linguistics*, 28(2):195–205.

Anoop Kunchukuttan. 2020. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

Pengfei Li, Liangyou Li, Meng Zhang, Minghao Wu, and Qun Liu. 2022. Universal conditional masked language pre-training for neural machine translation.

Ying Li and Pascale Fung. 2013. Improved mixed language speech recognition using asymmetric acoustic model and language model with code-switch inversion constraints. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7368–7372.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. 2020. Pre-training multilingual neural machine translation by leveraging alignment information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2649–2663, Online. Association for Computational Linguistics.

Yash Madhani, Sushane Parthan, Priyanka A. Bedekar, Ruchi Khapra, Vivek Seshadri, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2022. Aksharantar: Towards building open transliteration tools for the next billion users. *ArXiv*, abs/2205.03018.

Ravindra Nayak and Raviraj Joshi. 2022. L3Cube-HingCorpus and HingBERT: A code mixed hindi-english dataset and bert language models. *arXiv preprint arXiv:2204.08398*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 244–258, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Rana D Parshad, Suman Bhowmick, Vineeta Chand, Nitu Kumari, and Neha Sinha. 2016. What is india speaking? exploring the “hinglish” invasion. *Physica A: Statistical Mechanics and its Applications*, 449:375–389.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2021. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages.

Tapan Sahni, Chinmay Chandak, Naveen Reddy Chedeti, and Manish Singh. 2017. Efficient twitter sentiment classification using subjective distant supervision. In *9th International Conference on Communication Systems and Networks, COMSNETS 2017, Bengaluru, India, January 4-8, 2017*, pages 548–553. IEEE.

Pingali Sailaja. 2011. Hinglish: code-switching in indian english. *ELT J*, 65(4):473–480.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Anne-Marie Di Sciullo, Pieter Muysken, and Rajendra Singh. 1986. Government and code-mixing. *Journal of Linguistics*, 22(1):1–24.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W Black. 2019. A survey of code-switched speech and language processing. *arXiv preprint arXiv:1904.00784*.

Thamar Solorio, Shuguang Chen, Alan W. Black, Mona Diab, Sunayana Sitaram, Victor Soto, Emre Yilmaz, and Anirudh Srinivasan, editors. 2021. *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*. Association for Computational Linguistics, Online.

Vivek Srivastava and Mayank Singh. 2020. PHINC: A parallel Hinglish social media code-mixed corpus for machine translation. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 41–49, Online. Association for Computational Linguistics.

Vivek Srivastava and Mayank Singh. 2021. HinGE: A dataset for generation and evaluation of code-mixed Hinglish text. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 200–208, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jian Yang, Shuming Ma, Dongdong Zhang, ShuangZhi Wu, Zhoujun Li, and Ming Zhou. 2020a. Alternating language modeling for cross-lingual pre-training. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9386–9393.

Jian Yang, Yuwei Yin, Shuming Ma, Haoyang Huang, Dongdong Zhang, Zhoujun Li, and Furu Wei. 2021. Multilingual agreement for multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 233–239, Online. Association for Computational Linguistics.

Zhen Yang, Bojie Hu, Ambyera Han, Shen Huang, and Qi Ju. 2020b. CSP:code-switching pre-training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2624–2636, Online. Association for Computational Linguistics.

A XLM details

In order to backtranslate the Hinglish data, we hoped to train a good quality semi-supervised system using the XLM toolkit (Conneau et al., 2020b). We use Masked Language Modelling

(MLM) to pretrain a transformer encoder model on English, Hindi and Hinglish monolingual data. The model consisted of 6 layers, 1024 embedding dimensions, batch size 128, and a 0.1 dropout rate. We use 16.5M sentences of English WikiMatrix (Schwenk et al., 2021), 20M of HindiMono (Bojar et al., 2014) and 18.8M of Hinglish from L3Cube-HingCorpus (Nayak and Joshi, 2022) and CC-100 Hindi Romanized (Conneau et al., 2020a). Vocabulary and data preprocessing is the same as for the Marian models (4k BPE merges).

We then initialised a full transformer model with the pretrained encoder, and further trained with denoised autoencoding, MLM, machine translation[13], and backtranslation[14]objectives. We use the Samanantar corpus (10.1M) for the hi↔en translation objective, the 6.7k HinGe sentences as validation for hg↔en and hg↔hi directions, and the hi-en FLORES development set for hi↔en.

[13] hi↔en directions only

[14] Only direction involving hg: hi-hg-hi, en-hg-en, hg-hi-hg, hg-en-hg

CNLP-NITS-PP at MixMT 2022: Hinglish–English Code-Mixed Machine Translation

Sahinur Rahman Laskar[1], **Rahul Singh**[1], **Shyambabu Pandey**[1], **Riyanka Manna**[2]
Partha Pakray[1], **Sivaji Bandyopadhyay**[1]
[1]Department of Computer Science and Engineering, National Institute of Technology, Silchar, India
[2]Department of Computer Science and Engineering, Adamas University, Kolkata, India
{sahinurlaskar.nits, rahuljan, babushyampandey, riyankamanna16}@gmail.com
{parthapakray,sivaji.cse.ju}@gmail.com

Abstract

The mixing of two or more languages in speech or text is known as code-mixing. In this form of communication, users mix words and phrases from multiple languages. Code-mixing is very common in the context of Indian languages due to the presence of multilingual societies. The probability of the existence of code-mixed sentences in almost all Indian languages since in India English is the dominant language for social media textual communication platforms. We have participated in the WMT22 shared task of code-mixed machine translation with the team name: CNLP-NITS-PP. In this task, we have prepared a synthetic Hinglish–English parallel corpus using transliteration of original Hindi sentences to tackle the limitation of the parallel corpus, where, we mainly considered sentences that have named-entity (proper noun) from the available English-Hindi parallel corpus. With the addition of synthetic bi-text data to the original parallel corpus (train set), our transformer-based neural machine translation models have attained recall-oriented understudy for gisting evaluation (ROUGE-L) scores of 0.23815, 0.33729, and word error rate (WER) scores of 0.95458, 0.88451 at Sub-Task-1 (English-to-Hinglish) and Sub-Task-2 (Hinglish-to-English) for test set results respectively.

1 Introduction

The mixing of alternating words from two different language vocabulary without misinterpreting the context of the sentence is known as code-switching or code-mixing (Poulisse, 1998). This style of communication is one of the most frequent in multilingual communities, such as India. English is extensively mixed with local languages, such as Hindi, and Bengali, which causes code-mixed English-Hindi: Hinglish and English-Bengali: Binglish languages (Sailaja, 2011). Code-mixing is not observed in formal literature such as books but is commonly used on social media platforms such as Facebook and Twitter. The WMT22 organizes shared task code-mixed machine translation for English-to-Hinglish and Hinglish-to-English, where the main challenge is low-resource availability of parallel corpus. We have participated in the same task and to mitigate the issue of data scarcity, a synthetic Hinglish-English parallel corpus is prepared (as discussed in Section 3.1). In this work, the transformer-based neural machine translation (NMT) technique (Vaswani et al., 2017; Laskar et al., 2022) is utilized to build NMT models for both directions (English-to-Hinglish, Hinglish-to-English) of code-mixed MT.

2 Related Work

In recent times, many significant NLP studies have included the study of code-mixed languages. The EMNLP 2022 seventh conference on machine translation (WMT22) has put forward several tasks directed to meet new challenges in the field of NLP for code-mixed Indian languages. The competition has attracted many researchers to follow up with these tasks, which have eventually led to new directions and problems in this domain. The task of machine translation for code-mixed languages has not been an active area of research due to the scarcity of manually annotated datasets. Recently, researchers have been developing datasets for code-mixed MT that includes Hinglish-English parallel corpus, namely, HinGe (Srivastava and Singh, 2021) and PHINC (Srivastava and Singh, 2020) to overcome the datasets scarcity issue to build code-mix MT that is associated with the code-mixed text from various social media platforms. In this work, we addressed the issue of data scarcity by using synthetic Hinglish–English parallel corpus to increase the training data for code-mixed MT shared tasks at WMT22.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1158 - 1161
December 7-8, 2022 ©2022 Association for Computational Linguistics

3 System Description

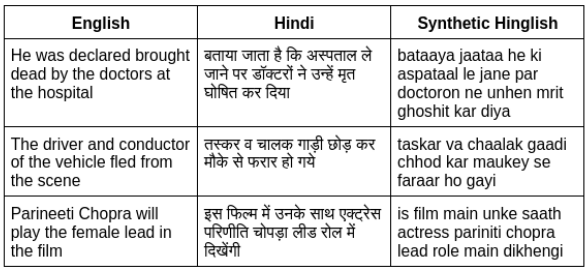
The experiments are carried out in four phases, namely, synthetic data preparation and augmentation to the train set, data preprocessing, model training, and testing. The OpenNMT-py (Klein et al., 2017) tool is utilized to build the NMT models independently for English-to-Hinglish (subtask-1) and Hinglish-to-English (subtask-2).

3.1 Dataset Description

We have used the dataset provided by the WMT22 organizer[1] and the statistics are presented in Table 1. Moreover, the synthetic English-Hinglish parallel dataset is prepared and directly augmented with the train set to expand the training amount of data. For synthetic data preparation, the English-Hindi parallel sentences are collected from Samanantar dataset (Ramesh et al., 2022) and selected $100k$ sentences (maximum length of 15 words). To select parallel sentences, the following steps are considered:

- Step-1: Extract proper nouns (named-entity) from the English side using NLTK[2] toolkit.
- Step-2: Extract English sentences that have extracted proper nouns in Step-1.
- Step-3: Select corresponding Hindi sentences of English that are extracted in Step-2.

Then, Hindi side sentences are transliterated into English script using Indic-trans[3] (Bhat et al., 2014) and prepared synthetic Hinglish sentences. Thus, we have prepared $100k$ Hinglish–English synthetic parallel corpus. The sample sentences of synthetic Hinglish-English are presented in Figure 1. The data statistics of the train set, before and after augmentation of synthetic Hinglish–English corpus is presented in Table 2.



English	Hindi	Synthetic Hinglish
He was declared brought dead by the doctors at the hospital	बताया जाता है कि अस्पताल ले जाने पर डॉक्टरों ने उन्हें मृत घोषित कर दिया	bataaya jaataa he ki aspataal le jane par doctoron ne unhen mrit ghoshit kar diya
The driver and conductor of the vehicle fled from the scene	तस्कर व चालक गाड़ी छोड़ कर मौके से फरार हो गये	taskar va chaalak gaadi chhod kar maukey se faraar ho gayi
Parineeti Chopra will play the female lead in the film	इस फिल्म में उनके साथ एक्ट्रेस परिणीति चोपड़ा लीड रोल में दिखेंगी	is film main unke saath actress pariniti chopra lead role main dikhengi

Figure 1: Sample sentences of synthetic Hinglish-English.

3.2 Experimental Setup

We have performed byte pair encoding jointly (subword level) (Sennrich et al., 2016) on the Hinglish-English with $32k$ merge operations. The subword level source-target vocabulary is shared during the training process of the NMT model. The OpenNMT-py toolkit has been used for text data tokenization, preprocessing, and conducting the NMT model training. We have followed the default settings of the 6 layer transformer model (Vaswani et al., 2017) in the training process. We have used a batch size of 32, 0.1 drop-outs, and an Adam optimizer with a 0.001 learning rate during the training process. The NMT model is trained on a single GPU with early stopping criteria, i.e., the model training is halted if it does not converge on the validation set for more than 10 epochs. The obtained trained model is used to translate the test data provided by the WMT22 organizers.

4 Results

The WMT22 shared task organizer published the evaluation result[4] of the code-mixed machine translation (MixMT) task for English–Hinglish language pair. We participated with the team name CNLP-NITS-PP in the monolingual to code-mixed machine translation: English-to-Hinglish (Sub-Task-1) and code-mixed to a monolingual machine translation: Hinglish-to-English (Sub-Task-2) submission tracks of the same task where ten teams participated. The automatic evaluation metrics, namely, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004), WER (Word Error Rate) (Morris et al., 2004) and human evaluation (HE) are used for the evaluation of results. Table 3, 4 reported the official results of our systems in terms of automatic and HE evaluation metrics. We

[1] https://www.statmt.org/wmt22/code-mixed-translation-task.html

[2] https://github.com/nltk/nltk

[3] https://github.com/libindic/indic-trans

[4] https://codalab.lisn.upsaclay.fr/competitions/2861#results

Task	Data Set	No. of Sentences	Tokens	
			English	Hinglish
Sub-Task-1	Train Set	2766	47347	52074
	Validation Set	500	5847	5565
	Test Set	1500	17694	17049
Sub-Task-2	Train Set	13738	169158	176410
	Validation Set	500	5847	10263
	Test Set	1500	27659	29335

Table 1: Data Statistics of English-Hinglish (provided by the organizer).

Train Set	Number of Parallel Sentence/Segments
Before Augmentation	2766 (Sub-Task-1) 13738 (Sub-Task-2)
After Augmentation	102,766 (Sub-Task-1) 113,738 (Sub-Task-2)

Table 2: Data Statistics of train set (before and after augmentation).

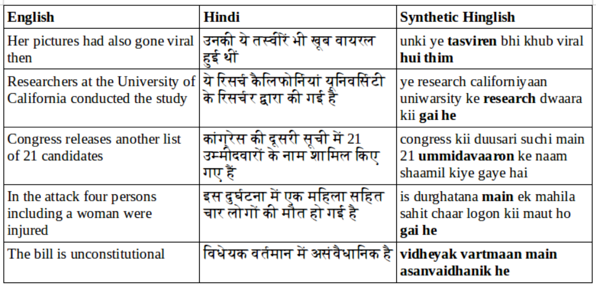
have attained better automatic evaluation scores and positions in Sub-Task-2 as compared to Sub-Task-1 for the validation and test set, whereas, in the case of human evaluation, we have achieved a higher score and position in Sub-Task-1 than Sub-Task-2. It is observed that due to the presence of a high amount of transliteration errors in synthetic code-mixed sentences, i.e., Hinglish, the predicted sentences suffer lower translation accuracy. A few examples of transliteration errors are presented in Figure 2.

Task	Set	ROUGE-L	WER
Sub-Task-1	Validation	0.23359 ($8th$)	0.97136 ($7th$)
	Test	0.23815 ($7th$)	0.95458 ($7th$)
Sub-Task-2	Validation	0.33835 ($4th$)	0.88002 ($3rd$)
	Test	0.33729 ($6th$)	0.88451 ($6th$)

Table 3: Our system's results (official) at MixMT shared task (WMT22).

Task	HE
Sub-Task-1	2.10 ($4th$)
Sub-Task-2	1.35 ($7th$)

Table 4: Our system's human evaluation results (official) at MixMT shared task (WMT22).



English	Hindi	Synthetic Hinglish
Her pictures had also gone viral then	उनकी ये तस्वीरें भी खूब वायरल हुई थीं	unki ye **tasviren** bhi khub viral **hui thim**
Researchers at the University of California conducted the study	ये रिसर्च कैलिफोर्नियां यूनिवर्सिटी के रिसर्चर द्वारा की गई है	ye research californiyaan uniwarsity ke **research** dwaara kii **gai he**
Congress releases another list of 21 candidates	कांग्रेस की दूसरी सूची में 21 उम्मीदवारों के नाम शामिल किए गए हैं	congress kii duusari suchi main 21 **ummidavaaron** ke naam shaamil kiye gaye hai
In the attack four persons including a woman were injured	इस दुर्घटना में एक महिला सहित चार लोगों की मौत हो गई है	is durghatana **main** ek mahila sahit chaar logon kii maut ho **gai he**
The bill is unconstitutional	विधेयक वर्तमान में असंवैधानिक है	**vidheyak vartmaan main asanvaidhanik he**

Figure 2: Sample examples of transliteration errors.

5 Conclusion and Future Work

In this work, we have investigated a transformer-based model for Hinglish–English language pair in the WMT22 code-mixed MT task. We have addressed the data scarcity issue by the augmentation of synthetic Hinglish–English parallel sentences to the train set for both English-to-Hinglish and Hinglish-to-English translation tasks (Sub-Task-1 and Sub-Task-2). Furthermore, synthetic parallel data will be corrected in the future to improve translational performance.

Acknowledgements

The authors would like to thank the Center for Natural Language Processing (CNLP), Artificial Intelligence (AI), and the Department of Computer Science and Engineering at National Institute of Technology, Silchar for providing the requisite support and infrastructure to execute this work.

References

Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tammewar, Riyaz Ahmad Bhat, and Manish Shrivastava. 2014. Iiit-h system submission for fire2014 shared task on transliterated search. In *Proceedings of the Forum for Information Retrieval Evaluation*, FIRE '14, page 48–53, New York, NY, USA. Association for Computing Machinery.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Sahinur Rahman Laskar, Abdullah Faiz Ur Rahman Khilji, Partha Pakray, and Sivaji Bandyopadhyay. 2022. Improved neural machine translation for low-resource english–assamese pair. *Journal of Intelligent and Fuzzy Systems*, 42(5):4727–4738.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Andrew Cameron Morris, Viktoria Maier, and Phil D. Green. 2004. From wer and ril to mer and wil: improved evaluation measures for connected speech recognition. In *INTERSPEECH*.

Nanda Poulisse. 1998. Duelling languages: Grammatical structure in codeswitching. *International Journal of Bilingualism*, 2(3):377–380.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Divyanshu Kakwani, Navneet Kumar, et al. 2022. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

Pingali Sailaja. 2011. Hinglish: code-switching in Indian English. *ELT Journal*, 65(4):473–480.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Vivek Srivastava and Mayank Singh. 2020. PHINC: A parallel hinglish social media code-mixed corpus for machine translation. In *Proceedings of the Sixth Workshop on Noisy User-generated Text, W-NUT@EMNLP 2020 Online, November 19, 2020*, pages 41–49. Association for Computational Linguistics.

Vivek Srivastava and Mayank Singh. 2021. Hinge: A dataset for generation and evaluation of code-mixed hinglish text. *CoRR*, abs/2107.03760.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Domain Curricula for Code-Switched MT at MixMT 2022

Lekan Raheem & Maab Elrashid
African Institute for Mathematical Sciences (AIMS)
{rwaliyu,mnimir}@aimsammi.org

Abstract

In multilingual colloquial settings, it is a habitual occurrence to compose expressions of text or speech containing tokens or phrases of different languages, a phenomenon popularly known as code-switching or code-mixing (CMX). We present our approach and results for the Code-mixed Machine Translation (MixMT) shared task at WMT 2022: the task consists of two subtasks, monolingual to code-mixed machine translation (Subtask-1) and code-mixed to monolingual machine translation (Subtask-2). Most non-synthetic code-mixed data are from social media but gathering a significant amount of this kind of data would be laborious and this form of data has more writing variation than other domains, so for both subtasks, we experimented with data schedules for out-of-domain data. We jointly learn multiple domains of text by pretraining and fine-tuning, combined with a sentence alignment objective. We found that switching between domains caused improved performance in the domains seen earliest during training, but depleted the performance on the remaining domains. A continuous training run with strategically dispensed data of different domains showed a significantly improved performance over fine-tuning.

1 Introduction

Code-mixing (CMX) denotes the alternation of two languages within a single utterance (Poplack, 1980; Sitaram et al., 2019). Code-mixing occurs mostly in unofficial groups in multilingual environments. More than 77% of Asians are multilingual (Ramakrishnan and Ahmad, 2014), and other statistics estimate that 64.5% of Europeans speak more than two languages, with more than 80% of adults in the region being bilingual (Eurostat, 2019). Code-mixing happens far more often in conversations than in writing, and mostly in unofficial settings, hence it rarely occurs in documented settings. This makes substantial data gathering for computational approaches to translations of code-mixed language difficult. Parallel corpora for code-switched data is very scarce (Menacer et al., 2019), this is because code-mixing mostly occurs in unofficial conversations like social media interactions.

Contemporary Neural Machine Translation (NMT) mostly makes use of parametric sequence-to-sequence models (Bahdanau et al., 2014; Vaswani et al., 2017), where an encoder receives a source sentence and outputs a set of hidden states, the decoder then scrutinizes these hidden states at each step, and outputs a sequence of softmax distribution over the target vocabulary space. Considering that we would need vast quantities of data to train an adequate NMT for this task, we leverage large-scale synthetic and available small data and notably rank data on domain relevance, by fine-tuning with it, initiating training with the relevant domain and strategically placing it at the premier batches of the training data.

Essentially, the characteristics of the data an NMT model is trained on are paramount to its translation quality, in particular in terms of size and domain. It is quintessential to train NMT models based on the domain relevance of corpora. Since most code-mixing occurs in unofficial communication, it is costly to find a lot of labeled data for every domain we are interested in. Hence we attempt to find less expensive exigencies to supplement training data, pretrain on largely available data of different domains, strategically construct synthetic data, and apportion data to make up for missing domains.

In these WMT Subtasks – monolingual to code-mixed machine translation (Subtask-1) and code-mixed to monolingual machine translation (Subtask-2), we also fine-tune on different domains, align representations of data and find the best combination of approaches to solving the subtasks. The main intuition behind our proposed solution is that NMT models exhibit a significant translation correlation when trained on data from the same or similar domains. With different data domain re-

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1162 - 1169
December 7-8, 2022 ©2022 Association for Computational Linguistics

quirements, it performs better when trained with data of the most relevant domain as preliminary batches compared to finetuning. As most natural code-mixed data source is social media and it is difficult to gather a good amount to train a model, it is incumbent to find a strategy that makes the model prioritize this form of data above others. Accordingly, we attempt to find less expensive techniques to supplement training data, pretrain on largely available data of different domains, strategically construct synthetic data, and apportion data to make up for missing domains. Our result showed improved performance on innate code-mixed data (and non-synthetic WMT test set samples) when this was prioritized and performed strongly in a test with a mix of several other data sources. We observed a better performance with domain-specific evaluation upon finetuning but this intensely plummeted performance on other 'pretraining domains', and more balanced performance on passing the interesting domain in the preliminary batches in a single 'all domain' training.

2 Related Work

It is laborious to obtain 'one-fits-all' training data for NMT. Most publicly available parallel corpora like Tanzil, OPUS, UNPC are sourced from documented communication, and these are often domain-specific. In NMT, data selection e.g. Axelrod et al. (2011) has remained as an underlying and important research concern. Choosing training examples that are relevant to the target domain, or by choosing high-quality examples for data cleaning (also known as denoising), has been essential in domain adaptation. Building a large-scale multi-domain NMT model that excels on several domains simultaneously becomes both technically difficult and practically back-breaking. Addressing research problems such as catastrophic forgetting (Goodfellow et al., 2013), data balancing (Wang et al., 2020), Adapters (Houlsby et al., 2019) have shown improvement. Unfortunately, several domains are difficult to handle with the single-domain data selection techniques currently in use. For instance, improving translation quality of one domain will often hurt that of another (Britz et al., 2017; van der Wees et al., 2017).

Song et al. (2019) replaced phrases with pre-specified translation to perform "soft" constraint decoding. Xu and Yvon (2021) generated code-mixed data from regular parallel texts and showed this training strategy yields MT systems that surpass multilingual systems for code-mixed texts.

Considering that code-mixed text belongs in less documented domains than most, there may be a need for domain adaptation used on sufficiently available data domains. Our work is inspired by the following approaches: Wang et al. (2019) executed simultaneous data selection across several domains by gradually focusing on multi-domain relevant and noise-reduced data batches while carefully introducing instance-level domain-relevance features and automatically constructing a training curriculum. Park et al. (2022) demonstrated that instance-level features are better able to distinguish between different domains compared to corpus-level attributes. Dou et al. (2019) proposed modeling the difference between domains instead of smoothing over domains for machine translation.

Anwar et al. (2022) showed that an encoder alignment objective is beneficial for code-mixed translation, in addition to Arivazhagan et al. (2019) that proposed auxiliary losses on the NMT encoder that imposed representational invariance across languages for multilingual machine translation.

English	Code-Mixed (CMX)
@dh*v*l2410*6 sure brother :)	@dh*v*l2410*6 sure bhai :)
"I just need reviews like these, this motivates me a lot"	"Bas aise hi reviews ki zaroorat hai, kaafi protsahan milta hai in baaton se. "
When the sorrow got missing in this room, the blood also became thin, #GuessTheSong	Jab gam ye rum mein kho gaya, toh khoon bhi patla hogaya #GuessTheSong

Table 1: Examples from the WMT Shared Task Dataset.

3 Data

In table 1 we show some samples from the WMT shared task, sourced from the non-synthetic validation data. The data provided for Subtask-1 (Srivastava and Singh, 2021) contains synthetic and human-generated data and Subtask-2 Parallel Hinglish Social Media Code-Mixed Corpus (Srivastava and Singh, 2020) for both tasks are mostly unofficial, mostly short conversational sentences, with some letters asterisked for privacy/derogatory reasons.

Since we need to augment provided data for a reasonable quantity to train a NMT model, we generated synthetic code-mixed data from the IITB

English	Code-Mixed (CMX)
Overhead charge is a percentage of the direct costs of providing the services under the contract.	Overhead charge, anubandh ke anusaar pradatt sevaon kee pratyaksh laagat ka ek pratishat hota hai.
A strategy of ignoring potential problems on the basis that they may be exceedingly rare.	us aadhaar par sambhaavit problems ko anadekha karane kee ek yukti, jahaan ki ve ati dushpraapy ho sakate hain.
A standard of measurement, or a unit that can be studied separately / independently.	koee maapadand athava koee a unit that svatantr roop se/alag se adhyayan kiya ja sakata ho.

Table 2: Examples from IITB Corpus.

corpus (Kunchukuttan et al., 2017) which is from 17 sources of different domain mostly HindEnCorp (Bojar et al., 2014), Gyaan-Nidhi Corpus (Garg et al., 2018), Indian Government corpora - CFILT, Mahashabdkosh, Tanzil, and GNOME (Kunchukuttan et al., 2017) (details in section 3.1). Synthetic code-switched sentences generated from the IITB corpus belong to a different domain than the WMT evaluation data, as we illustrate with the English translation samples in table 2.

For the pretraining-finetuning setup, we pretrain with synthetic code-switched data generated from IITB corpus and fine-tune on the WMT data provided for each task. For both pretraining and finetuning, we coordinate the data similar to (Anwar et al., 2022) – For Subtask-1, Monolingual to code-mixed machine translation subtask, we use the Hindi sentence (Devanagari script) as source sequence and the corresponding code-switched sentence (Roman script) as target, then alternated the English sequence (Roman script) as source sentence and the same corresponding code-switched sentence as the target sequence. The above two source-target parallel data are set after each other. For Subtask-2, Code-mixed to monolingual machine translation subtask, we have a similar arrangement as in Subtask-1, but with the source sequences of Hindi and code-mixing (Hinglish) in Roman script and as the target the corresponding English sequence. We removed sequences shorter than 2 tokens, and those longer than 250 tokens, and a target-to-source token ratio of more than 1.5. After cleaning the pretraining data, for Subtask-1, we have about 2.5M parallel sentences and 2.3M parallel sentences for Subtask-2.

For the finetuning process, we made use of the WMT training data provided for each subtask and organized like the pretraining data as described above. After cleaning, for Subtask-1 (Synthetic + Human-generated), we have a total of over 11K parallel sentences. For subtask-2, over 12K parallel sentences remain after cleaning.

Since the IITB corpus encompasses multiple sources and domains where code-mixing infrequently occurs, we decided to configure our model in a way it first learns from natural code-mixed data provided by WMT. We experiment with a hand-designed curriculum of the Synthetic Code-switched data generated from the IITB corpus and the WMT provided data. We supply the model the non-synthetic WMT data only in the first few batches in the hope that this would faintly familiarize the model with domain-specific features before it learns from the synthetic code-switched data we generated from other domains. We compare the results of this approach to the above described pretraining-finetuning setup. All data is tokenized and normalized using sentencepiece[1].

3.1 Code Switched Data Generation

Given that most publicly available corpora are monolingual, it is requisite to generate sufficient synthetic code-mixed data for training. Moreover, there have been works on generating synthetic code-mixed data linguistically, there are a few rules theories that are essential.

The **Equivalence Constraint Theory** states that intra-sentential code-mixing can only happen where the surface structures of two languages map onto each other, implicitly following both languages' grammatical norms (Poplack, 1980). Fundamentally, we can only attempt code-mixing at points where both languages coincide on the parse tree to equivalent phrase structure.

The **Matrix Language Theory** explains code-mixing by introducing the concept of a "Matrix Language," or base language, into which clusters of the "Embedded Language," or second language, are introduced in such a way that the former sets the grammatical structure of the sentence and the latter "switches-in” at grammatically correct points of the sentence (Myers-Scotton, 2001). The Matrix language has more tokens in the sequence and its rules are designated above the embedded language's.

Considering the linguistic theories above, we

[1] https://github.com/google/sentencepiece

generate code-mixed data by locating where both languages coincide based on a word-level alignment extracted and only replace tokens based on the "matrix language theory". Roughly following the recipes by (Song et al., 2019; Rizvi et al., 2021; Xu and Yvon, 2021; Anwar et al., 2022), we generate synthetic code-switched data from the IITB parallel data: We create code-mixed data by first transliterating Hindi (Devanagari script) to Roman script using Ritwik's tool[2], then extract word alignments using the giza++ toolkit (Och and Ney, 2003), and extract minimal alignment units following the approach of (Crego, 2005). We choose Hindi as the "matrix language" by determining this from the provided WMT training data, we extract word alignments and find how many tokens in each sequence belongs to which language using the language detector of Googletrans python library[3] and assign the language with more tokens as the matrix language. Figure 1 shows the Hindi/English matrix language ratio for both subtasks.

Similar to MLM pre-training used by BERT (Devlin et al., 2018), we randomly replace 15% of the tokens in each Hindi sentence with their aligned segments in the embedded language (English). For short sequences with less than 7 tokens we make only one replacement, chosen uniformly at random.

4 Training Objective

Considering the effectiveness of clean finetuning (Wu et al., 2019), and pre-training (Mathis et al., 2019), we attempt a combined pipeline of pretraining+finetuning experiment and also a single training but with tactical positioning of the most important domain. In the finetuning process and training with specially ordered data, as recommended by (Anwar et al., 2022), we add an alignment loss to the encoder to encourage source and target representations to be close in representation space minimizing the max-pooled cosine distance of the encoder representation as shown in equation 1:

$$\Omega = \mathbb{E}_{D_{(en,hi)}}[1 - sim(Enc(x_{src}), Enc(x_{tgt}))] \quad (1)$$

Where Ω is the encoder loss, $D_{(en,hi)}$ is the data consisting of the parallel pairs of code-mixed to monolingual or monolingual to code mixed depending on which subtask the data belongs to, x_{src} is the source sequence and x_{tgt} is the target sequence, Enc(x) is the max-pooled encoder representation of sentence x similar to (Gouws et al., 2014) and (Coulmance et al., 2015), and sim is the cosine similarity. Unlike (Arivazhagan et al., 2019) where the whole model's parameters are updated as shown in figure 2.

5 Experiments and Results

In all of our experiments, we used Transformer-Base (Vaswani et al., 2017) configuration with the Fairseq (Ott et al., 2019) framework. All models were trained on four Tesla T400 GPUs using IITB and WMT-'22 MixMT data for training as described in Section 3, with a shared vocabulary of 77K BPE (Sennrich et al., 2015) sub-words to create a joint vocabulary for both tasks and all models. The model's hyperparameters can be found in Appendix A.

5.1 Results

Based on the human evaluation by the organizers of the subtasks, the translation result of our initial models - v0.2 submitted – which was trained with mixing the IITB with the WMT without prioritizing the target domain – had an overall rating of 1.75 from 10 random translations for each subtask, this ranked inferior to many other submissions.

With the help of native Hindi speakers to investigate our data, we found some of the causes it performed decumbent, which were as a result of some of the different data preprocessing tools we used: For Transliteration, We tried a few devanagari to roman tools but had some shortcomings like:

- Lipika-ime[4]: inappropriate handling of diacritic characters.
- Indic-trans[5]: Removal of vowels (e.g. default -> difolt, highlight -> hilite, method -> methd, etc..), Splitting of words that lead to suboptimal outputs (e.g. "un he" instead of "unhe").
- Sheental[6]: repetition of vowels e.g. jane -> jaane, yaar -> yaara, incorrect replacement of characters e.g. om -> on and occurence of

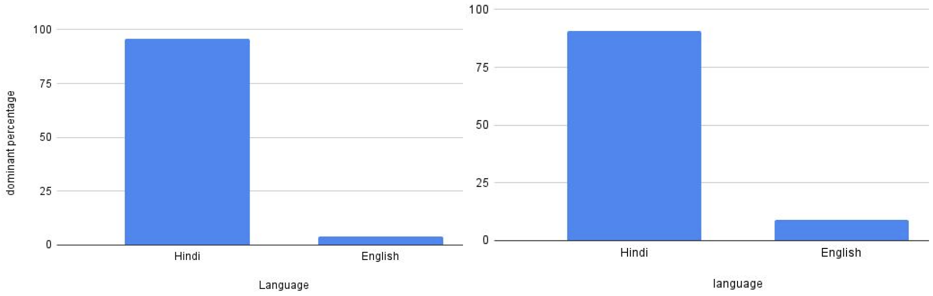
[2] https://github.com/ritwikmishra/devanagari-to-roman-script-transliteration

[3] https://github.com/ssut/py-googletrans/blob/master/docs/index.rst

[4] https://github.com/ratreya/lipika-ime

[5] https://github.com/libindic/indic-trans

[6] https://github.com/sheetalgiri/devanagari-to-roman-script



(a) Subtask-1

(b) Subtask-2

Figure 1: Percentage of Hindi vs. English as matrix language from WMT'22 Hinglish validation data for the subtasks.

Model	**IITB Eval Set**		**WMT Eval Set**		**Mixed Eval Set**	
	Subtask-1	Subtask-2	Subtask-1	Subtask-2	Subtask-1	Subtask-2
Pretrained (IITB corpus only)	**0.81**	**0.85**	0.41	0.47	**0.76**	**0.80**
Pretrained (IITB corpus) + Finetuned (WMT provided)	0.49	0.52	0.54	0.58	0.53	0.59
Mixed-data training (target domain first)	0.76	0.79	**0.62**	**0.64**	0.70	0.73

Table 3: Translation accuracy of subtask-1 and subtask-2 of Hindi-English in ROUGE-L (F1-Score) on different **test** data of different domains, based on models trained on different domain training data, data arrangement or training pipeline.

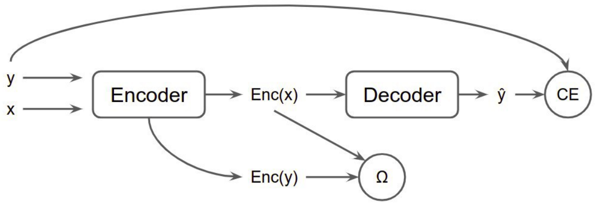


Figure 2: The loss function visualization, CE is the Cross Entropy, Ω is the encoder loss.

needless suffixes e.g. palat -> palata, some diacritic appeared independently.

- Ritwik's: inappropriately breaking very long sentences into multiple lines, replacing individually occurring tokens like um -> oon and abruptly stopping when ran over large amount of data so we divided the data into chunks each containing not more than 200K sequences, optimized by parallel computing using dask, and replaced the individually occurring tokens changed afterwards.

We also investigated our initial model and discovered a few other issues like:

- Cases of translation of proper nouns in Subtask-2 (e.g. Sapna -> dream) which we deduce as a pointer to insufficient training data.
- Imprecise tokenization and detokenization, we also switched to use of Google sentencepiece instead of Moses SMT
- Also, the organizers noticed the team's output had an incorrect order. A problem where the post-processing had sorted the hypothesis and fragmented longer sentences also influenced the rating.

Upon inspecting our model outputs we found a few inaccuracies with the tools we used for transliteration and tokenization for the submitted model hypotheses. We fixed these, and present the results in the following section.

5.2 Post-Submission Results

Table 3 shows the experimental results based on different test data of samples each from IITB corpus, WMT, and a Mixed test sample evenly selected from Samanatar (includes IITB corpus, CCMatrix, Hindi-News, Jagran, Livehindustan, Patrika and WMT). We made use of other reputable tools to fix the aforementioned errors, added the domain curriculum technique, and ran the experiment again and present it in table 3.

Table 3 shows that fine-tuning on the WMT domain improves translation accuracy on this domain slightly, but the model suffers 'catastrophic forgetting' on domains it was initially trained on. Pretraining did not lead to a good generalization for the WMT test samples provided, hence a need for domain adaptation. Placing the relevant domain in the preliminary batches for mixed-data training also improves training on such a domain but hurts other domains slightly.

6 Conclusion

We present a data domain sorting method that improves translation performance based on a target domain for the WMT 2022 code-switching shared tasks. We compared our result to a pretraining and fine-tuning pipeline, and demonstrated that the finetuning method improves on specified domain but upsets on previously learned data domain. An aspect we intend to delve further into is efficient domain adaptation strategies that may help low-resource domains such as code-mixing, and have little or no effect on high-resource domains, we are currently looking into domain adaptation learning curve (Park et al., 2022), extraction of domain-specific parameters (Dou et al., 2019) for better data augmentation strategies, better acquisition of code-mixed data, and the use of Adapters (Houlsby et al., 2019).

Acknowledgement

We would like to express special gratitude to Julia Kreutzer and Melvin Johnson of Google Research, Mohamed Anwar and Sweta Agrawal for their support and contributions. We also appreciate the GCP credits granted by Google for this project.

References

Mohamed Anwar, Lekan Raheem, Maab Elrasheed, Melvin Johnson, and Julia Kreutzer. 2022. True bilingual NMT. In *3rd Workshop on African Natural Language Processing*.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Roee Aharoni, Melvin Johnson, and Wolfgang Macherey. 2019. The missing ingredient in zero-shot neural machine translation. *CoRR*, abs/1903.07091.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate.

Ondřej Bojar, Vojtěch Diatka, Pavel Rychlý, Pavel Straňák, Vít Suchomel, Aleš Tamchyna, and Daniel Zeman. 2014. HindEnCorp - Hindi-English and Hindi-only corpus for machine translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3550–3555, Reykjavik, Iceland. European Language Resources Association (ELRA).

Denny Britz, Quoc Le, and Reid Pryzant. 2017. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 118–126, Copenhagen, Denmark. Association for Computational Linguistics.

Jocelyn Coulmance, Jean-Marc Marty, Guillaume Wenzek, and Amine Benhalloum. 2015. Trans-gram, fast cross-lingual word-embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1109–1113, Lisbon, Portugal. Association for Computational Linguistics.

Josep Crego. 2005. Reordered search and tuple unfolding for ngram-based smt.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Zi-Yi Dou, Xinyi Wang, Junjie Hu, and Graham Neubig. 2019. Domain differential adaptation for neural machine translation. *CoRR*, abs/1910.02555.

Eurostat. 2019. Translate foreign language skills statistics. `https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Foreign_language_skills_statistics#Number_of_foreign_languages_known`.

Kamal Garg, Ajit Kumar, and Vishal Goyal. 2018. Development of punjabi-english (puneng) parallel corpus for machine translation system. pages 690–693.

Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2014. Bilbowa: Fast bilingual distributed representations without word alignments.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2017. The IIT bombay english-hindi parallel corpus. *CoRR*, abs/1710.02855.

Alexander Mathis, Mert Yüksekgönül, Byron Rogers, Matthias Bethge, and Mackenzie W. Mathis. 2019. Pretraining boosts out-of-domain robustness for pose estimation. *CoRR*, abs/1909.11229.

Mohamed Menacer, David Langlois, Denis Jouvet, Dominique Fohr, Odile Mella, and Kamel Smaïli. 2019. Machine Translation on a parallel Code-Switched Corpus. In *Canadian AI 2019 - 32nd Conference on Canadian Artificial Intelligence*, Lecture Notes in Artificial Intelligence, Ontario, Canada.

Carol Myers-Scotton. 2001. The matrix language frame model: Development and responses. *Trends in Linguistics Studies and Monographs*, 126:23–58.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *CoRR*, abs/1904.01038.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation.

Cheonbok Park, Hantae Kim, Ioan Calapodescu, Hyunchang Cho, and Vassilina Nikoulina. 2022. Dalc: Domain adaptation learning curve prediction for neural machine translation.

Shana Poplack. 1980. Sometimes i'll start a sentence in spanish y termino en espaÑol: toward a typology of code-switching 1. *Linguistics*, 18:581–618.

Karthick Ramakrishnan and Farah Z. Ahmad. 2014. Language diversity and english proficiency part of the "state of asian americans and pacific islanders" series.

Mohd Sanad Zaki Rizvi, Anirudh Srinivasan, Tanuja Ganu, Monojit Choudhury, and Sunayana Sitaram. 2021. GCM: A toolkit for generating synthetic code-mixed text. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 205–211, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W. Black. 2019. A survey of code-switched speech and language processing. *CoRR*, abs/1904.00784.

Marina Sokolova, Nathalie Japkowicz, and Stan Szpakowicz. 2006. Beyond accuracy, f-score and roc: A family of discriminant measures for performance evaluation. volume Vol. 4304, pages 1015–1021.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified translation. *CoRR*, abs/1904.09107.

Vivek Srivastava and Mayank Singh. 2020. Phinc: A parallel hinglish social media code-mixed corpus for machine translation. *arXiv preprint arXiv:2004.09447*.

Vivek Srivastava and Mayank Singh. 2021. Hinge: A dataset for generation and evaluation of code-mixed hinglish text. *arXiv preprint arXiv:2107.03760*.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Wei Wang, Ye Tian, Jiquan Ngiam, Yinfei Yang, Isaac Caswell, and Zarana Parekh. 2019. Learning a multitask curriculum for neural machine translation. *CoRR*, abs/1908.10940.

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020. Balancing training for multilingual neural machine translation. *CoRR*, abs/2004.06748.

Felix Wu, Angela Fan, Alexei Baevski, Yann N. Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. *CoRR*, abs/1901.10430.

Jitao Xu and François Yvon. 2021. Can you traducir this? machine translation for code-switched input. *CoRR*, abs/2105.04846.

A Appendix

Table 4 holds all the hyper-parameters we used for training all models. All experiments were set to halt at patience of 15 updates on the BLEU (Papineni et al., 2002) stabilizing, we found it trained longer with BLEU, but evaluated on WMT specified F1-Score (Sokolova et al., 2006) for the subtasks.

Hyper-parameter	Value
Number of Layers	6
Hidden size	512
FFN inner hidden size	2048
Attention heads	8
Attention head size	64
Dropout	0.1
Attention Dropout	0.0
Warmup Steps	4000
Learning Rate	5e-4
Learning Rate Decay	inverse_sqrt
Batch Size	4096 tokens
Label Smoothing	0.1
Weight Decay	0.0001
Adam ϵ	10^{-9}
Adam β_1	0.9
Adam β_2	0.98
Encoder Criterion Weight	10

Table 4: The hyperparameter values setting for training.

Lingua Custodia's participation at the WMT 2022 Word-Level Auto-completion shared task

Melissa Ailem, Jinghsu Liu, Jean-Gabriel Barthélemy and Raheel Qader
Lingua Custodia, France
{melissa.ailem,jingshu.liu,j-g.barthelemy,raheel.qader}@linguacustodia.com

Abstract

This paper presents Lingua Custodia's submission to the WMT22 shared task on Word Level Auto-completion (WLAC). We consider two directions, namely German-English and English-German. The WLAC task in Neural Machine Translation (NMT) consists in predicting a target word given few human typed characters, the source sentence to translate, as well as some translation context. Inspired by recent work in terminology control, we propose to treat the human typed sequence as a constraint to predict the right word starting by the latter. To do so, the source side of the training data is augmented with both the constraints and the translation context. In addition, following new advances in WLAC, we use a joint optimization strategy taking into account several types of translation context. The automatic as well as human accuracy obtained with the submitted systems show the effectiveness of the proposed method.

1 Introduction

Modern advances in Neural Machine Translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2014; Luong et al., 2015; Vaswani et al., 2017) gave rise to a new era, where the translation quality significantly surpasses previous statistical machine translation (SMT) models (Och and Ney, 2002; Koehn et al., 2003; Koehn, 2010).

Although these approaches generate high quality translations, there is still a long way to go towards meeting human quality. In fact, NMT models can still generate several types of grammatical and/or semantic mistakes, which is not tolerated in scenarios requiring accurate and prompt translations. These scenarios include for instance the translations of legal and financial documents, where mistakes are not permitted and can be costly. To overcome this issue, several Computer-aided translation (CAT) systems have been proposed (Knowles and Koehn, 2016; Santy et al., 2019) to refine NMT models. CAT tools include for instance Automatic Post-edition (Junczys-Dowmunt and Grundkiewicz, 2017; Correia and Martins, 2019; Lopes et al., 2019), terminology control (Hokamp and Liu, 2017; Post and Vilar, 2018; Dinu et al., 2019; Ailem et al., 2021) and sentence vs word-level autocompletion (Knowles and Koehn, 2016; Zhao et al., 2020; Li et al., 2021).

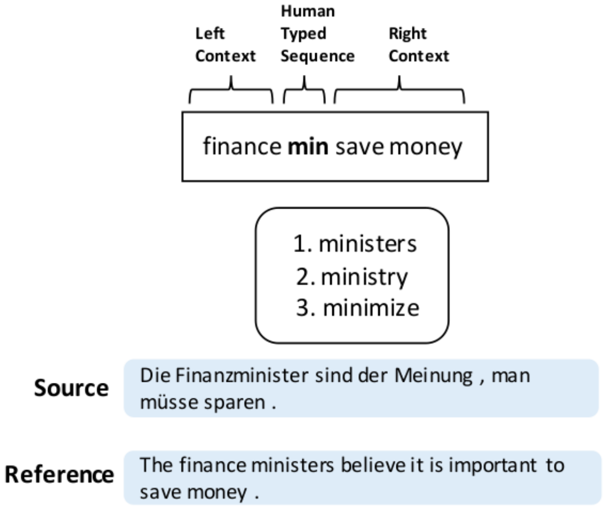

Left Context
Human Typed Sequence
Right Context
finance **min** save money
1. ministers
2. ministry
3. minimize
Source
Die Finanzminister sind der Meinung , man müsse sparen .
Reference
The finance ministers believe it is important to save money .

Figure 1: Given the source sentence to translate, the translation contexts, and the human typed characters, the WLAC task aims to predict a target word starting by the human typed sequence. As illustrated, the word to predict is not necessarily consecutive to the left and right contexts.

The current shared task is on Word-Level Auto-Completion (WLAC) methods, whose objective, as illustrated in Figure 1, is to predict a target word given a source sentence, a translation context, and at least one human typed character. WLAC is a central Computer-aided task as it helps human translators generate diverse translations quickly and effectively. Unfortunately, due to the lack of benchmark datasets, very little work has considered this task. Existing methods include the work of Huang et al. (2015), where the authors leverage the source sen-

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1170 - 1175
December 7-8, 2022 ©2022 Association for Computational Linguistics

tence as well as human typed characters to predict the target word. More recently, Li et al. (2021) proposed to use context information in addition to human-typed characters and source sentence. Furthermore, the authors presented a generic procedure to simulate WLAC data from any parallel translation datasets, and proposed the first public benchmark for this task. The benchmark dataset contains several types of contexts and therefore a joint optimization strategy is used to take into account all context types during training.

We participate in two directions, namely English-German (EN-DE), German-English (DE-EN), and we submitted four systems, two for each language direction. Following previous work (Li et al., 2021), our method leverages source sentence, translation context as well as human-typed characters, and it uses a joint objective function to learn model parameters on different types of contexts simultaneously. Furthermore, inspired by recent progress in Terminology Control (TC) for NMT (Dinu et al., 2019; Ailem et al., 2021), we propose a new WLAC method that treats the human typed sequence as a constraint to generate the right word. To do so, we augment our training data with translation context as well as human typed characters (constraints). We use tags where needed to distinguish these terms from source tokens.

The rest of the paper is organized as follows. Section 2 describes the details of our system, section 3 presents the data, while section 4 shows the different experimental settings and results.

2 Method

Herein we present our WLAC approach which is inspired by recent advances in this task (Li et al., 2021) as well as recent work on terminology control (Ailem et al., 2021).

2.1 Data Annotation

Inspired by previous work on Terminology Control (Ailem et al., 2021), the idea here is to consider human typed characters as a constraint. In particular, the objective is to constrain the NMT model to predict a word that, obligatorily, starts with human typed characters. To do so, we augment the source side of our training data with the translation context as well as the human typed sequence of characters. Furthermore, we use tags to specify the constraints (human typed characters) in the context translation where relevant, and use the special token MASK in order to provide a more general pattern for the model to learn how to predict the right word starting with human sequence. The WLAC data provided by the WMT task and used in (Li et al., 2021) contains 4 types of context, namely left and right contexts (bi-context), left context only (prefix), right context only (suffix), and no context at all (zero context). The different annotations according to each context types are depicted in table 1.

2.2 Joint Cross-Entropy Loss

Let $\mathbf{x} = (x_1, x_2, ..., x_m)$ denotes the input sentence to translate, $\mathbf{s} = (s_1, s_2, \ldots, s_k)$ a sequence of human typed characters, and $\mathbf{c} = (c_l, c_r)$ the translation context, where $c_l = (c_{l,1}, c_{l,2}, \ldots, c_{l,i})$ denotes the left context, while $c_r = (c_{r,1}, c_{r,2}, \ldots, c_{r,j})$ denotes the right context. The objective of the WLAC task is to predict a word w given a source sequence $\mathbf{x}$, human typed sequence $\mathbf{s}$ and a translation context $\mathbf{c}$ in order to establish a partial translation. The training data $\mathcal{D}$ of a WLAC task can be described as a set of tuples $(\mathbf{x}, \mathbf{s}, \mathbf{c}, w)$. From a probabilistic perspective, a WLAC task can be cast as estimating the conditional distribution $p(w|\mathbf{x}, \mathbf{c}, \mathbf{s})$. Since there is different types of context (as described in section 2.1), we follow the work of Li et al. (2021) and adopt a joint training strategy. In particular, the four types of context are considered during training giving rise to the following loss function:

$$\begin{aligned}\mathcal{L} &= -\log p(w|\mathbf{x}, \mathbf{c}, \mathbf{s}) \\ &= -\sum_{(\mathbf{x},\mathbf{c},\mathbf{s},w)\in\mathcal{D}_{bi}} \log p(w|\mathbf{x}, c_l, c_r, \mathbf{s}) \\ &\quad - \sum_{(\mathbf{x},c_r,\mathbf{s},w)\in\mathcal{D}_{suf}} \log p(w|\mathbf{x}, c_r, \mathbf{s}) \\ &\quad - \sum_{(\mathbf{x},c_l,\mathbf{s},w)\in\mathcal{D}_{pre}} \log p(w|\mathbf{x}, c_l, \mathbf{s}) \\ &\quad - \sum_{(\mathbf{x},\mathbf{s},w)\in\mathcal{D}_{zero}} \log p(w|\mathbf{x}, \mathbf{s})\end{aligned} \tag{1}$$

where $\mathcal{D}_{bi}$, $\mathcal{D}_{suf}$, $\mathcal{D}_{pre}$, $\mathcal{D}_{zero}$ correspond respectively to bi-context, suffix context, prefix context and zero context.

3 Data

We participate in two directions, namely English-German and German-English. We use the parallel

Source	Seebarsch gebacken auf seinem Rücken , fein zerschnippelter Porree und Zitronenmelissekraut .			
Target	Bar baked on the back with finely chopped leek and lemon melissa herbs .			
WLAC training data				
Input	bi-context	Seebarsch gebacken auf seinem Rücken , fein zerschnippelter Porree und Zitronenmelissekraut . <SEP> Bar baked on <S> MASK <C> wit </C> and lemon	Output	with
	Prefix Context	Seebarsch gebacken auf seinem Rücken , fein zerschnippelter Porree und Zitronenmelissekraut . <SEP> Bar baked on the back with finely <S> MASK <C> chop </C>		chopped
	Suffix Context	Seebarsch gebacken auf seinem Rücken , fein zerschnippelter Porree und Zitronenmelissekraut . <SEP> <S> MASK <C> B </C> lemon melissa herbs .		Bar
	Zero Context	Seebarsch gebacken auf seinem Rücken , fein zerschnippelter Porree und Zitronenmelissekraut . <SEP> <S> MASK <C> bak </C>		baked

Table 1: Illustration of our German-English training data. The WLAC training data can be build from traditional parallel translation data. During sampling, for each parallel sentences four samples are generated corresponding to four types of translation context, namely left and right contexts (bi-context), left context (prefix), right context (suffix) and no context at all (zero). The source side of the training data is a concatenation of the German source side and the English translation context separated by the tag **<SEP>**. Translation context is also augmented with human typed characters, which are considered as a constraint to orient the model to predict the right word. The tags <S>, <C> and </C> are added to differentiate between the constraints and other tokens in the input.

English-German data provided by the WLAC task consisting of almost 4.5M parallel sentences. Following task instructions, we use the script proposed in (Li et al., 2021) to simulate the WLAC training data from the provided classical translation data.

3.1 Parallel Data Cleaning

Before creating the WLAC samples, we apply several cleaning steps on the data to eliminate bad alignments. First, the data is re-segmented using the Python package pySBD (Sadvilkar and Neumann, 2020) in order to detect sentence boundaries. This step increases the number of parallel sentences to almost 6.5 M. Second, each parallel entry is scored between 0 and 1 using several tools. These tools include bicleaner (Ramírez-Sánchez et al., 2020), similarity scoring using LaBSE (Feng et al., 2020), and bicleaner-ai, which is inspired by the BERT-based model proposed in (Açarçiçek et al., 2020) for sentence classification. Table 2 presents the different scoring thresholds used to clean the parallel corpus. In particular, we rely on a combination of bi-cleaner and similarity scoring as well as bicleaner-ai. In our experiments, we consider both initial uncleaned data (Noisy) and the cleaned data. In addition to these scoring, we also rely on fast-text (Bojanowski et al., 2017) to eliminate sentence pairs identified as written in the wrong language (e.g., A french sentence in an English-German parallel corpus). After the cleaning we obtain a corpus of around 2.7 M parallel segments.

	Clean	Noisy
Bicleaner + Similarity	>1.4	>0
Bicleaner-ai	>0.25	>0.1
Total sentences	2 717 737	4 404 427

Table 2: The different thresholds applied on the corpus. Noisy corresponds to the original parallel data provided by the WLAC task. The threshold 1.4 is a combination of Bi-cleaner and Similarity scoring thresholds.

3.2 WLAC Data Construction

The parallel data commonly used for NMT and provided by the WLAC task cannot be used directly to train a WLAC model. Thus, following the task instructions, we use the script proposed in (Li et al., 2021) to simulate several samples for the WLAC training. For each sentence pair, 4 samples are created according to the four context types as presented in table 1. Since the provided initial data contains almost 4.5M parallel sentences, we obtain a WLAC corpus of almost 18M entries (4.5×4). As presented in the previous section, we have also used a cleaned version of the provided corpus, containing around 2.7M entries. For the

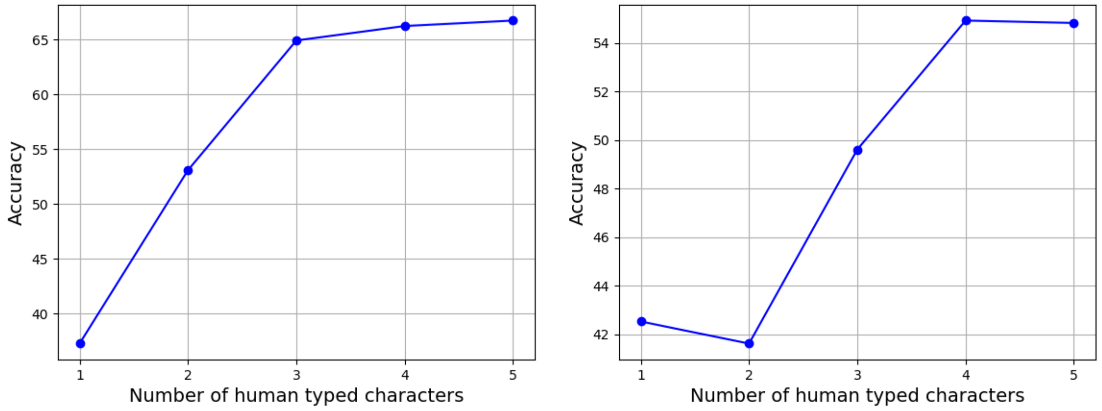


Figure 2: Accuracy obtained with different number of human typed characters. Left : German-English system with initial corpus. Right : English-German system with initial corpus.

latter, we obtain around 10.8M WLAC training samples. Hence, synthetic WLAC training data are build for the two corpus versions (clean and initial) in the two considered directions: English-German and German-English. The dev sets are build from 3000 EN-DE and DE-EN parallel sentences from the initial corpus. To do so, the same sampling script is used resulting in 20K entries for both directions. The test sets released by the WLAC task contain 29596 and 25895 samples for DE-EN and EN-DE respectively.

4 Experiments

4.1 Settings

We use a Transformer architecture (Vaswani et al., 2017) with 6 stacked encoders/decoders and 8 attention heads as a building block for our systems. For both EN-DE and DE-EN, the source and target embeddings are tied with the softmax layer. We use 512-dimensional embeddings, 2048-dimensional inner layers for the fully connected feed-forward network and a dropout rate of 0.3. The models are trained for a minimum of 50 epochs and the validation set is used to compute the stopping criterion[1]. We use a batch size of 4000 tokens per iteration and an initial learning rate of 5×10^{-4}. For each language pair, the validation set is used to compute the stopping criterion. We use a beam size of 5 during inference for all models.

Before annotating our corpus as presented in table 1, we first tokenize the data using Moses tokenizer (Koehn et al., 2007). After augmenting the data with translation context and human typed sequence, we perform a BPE encoding (Sennrich et al., 2015) with 40k merge operations to segment words into subword-units, which results in a joint vocabulary size of around 44K tokens for both German-English and English-German.

Accuracy (%)		
	German-English	English-German
Cleaned Corpus	54.84	48.43
Initial Corpus	57.36	48.97
Human Evaluation (%)		
Cleaned Corpus	74.50	61.00
Initial Corpus	76.75	61.75

Table 3: Accuracy and Human Evaluation results.

4.2 Results

For both considered directions, the systems are evaluated using the Accuracy measure, corresponding to the percentage of correctly predicted words. This automatic accuracy is obtained using one single ground truth word for each sample. However, one sentence may have multiple translations, thus several Ground Truth are possible, making the automatic accuracy inadequate. To overcome this limitation, a human evaluation is applied on 400 randomly sampled entries from the test set. In particular, given the human typed sequence, the translation context and the source sentence to trans-

[1] The stopping criterion corresponds to 5 successive epochs without decreasing the validation loss function.

late, human annotators judge whether a predicted word can be correct according to the given context. The results obtained with our systems are presented in table 3.

Surprisingly, we observe that cleaning the different corpora is mirrored by a deterioration in results. Indeed, the best results are reached with the systems using initial training corpus. This might be due to the excessive cleaning, removing some scenarios that could be present in the test set.

Furthermore, we notice that the chances of predicting the right word are positively related with the number of human typed characters. We present in figure 2 the accuracy obtained with different numbers of human typed characters. In both directions, we observe that the accuracy improves with the typed sequence length. This is natural, as with few typed characters, several choices are possible, especially when the translation context is restricted or even non-existent (zero context situation).

5 Conclusion

This paper describes our submission to the WLAC shared task. We participate in two language directions, EN-DE and DE-EN, and submitted two systems for each direction. For each direction, the first system is trained using initial data provided by the task, while the second system is trained on cleaned data. The evaluation in terms of accuracy shows the effectiveness of the proposed method. Furthermore, a significant improvement of accuracy is observed when the number of human typed characters is greater than 1, suggesting that entering at least two characters restrain the search space and improve the chances to predict the right word.

References

Haluk Açarçiçek, Talha Çolakoğlu, Pınar Ece Aktan Hatipoğlu, Chong Hsuan Huang, and Wei Peng. 2020. Filtering noisy parallel corpus using transformers with proxy task learning. In *Proceedings of the Fifth Conference on Machine Translation*, pages 940–946, Online. Association for Computational Linguistics.

Melissa Ailem, Jinghsu Liu, and Raheel Qader. 2021. Encouraging neural machine translation to satisfy terminology constraints. *arXiv preprint arXiv:2106.03730*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Gonçalo M Correia and André FT Martins. 2019. A simple and effective approach to automatic post-editing with transfer learning. *arXiv preprint arXiv:1906.06253*.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, page 3063–3068.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, page 1535–1546.

Guoping Huang, Jiajun Zhang, Yu Zhou, and Chengqing Zong. 2015. A new input method for human translators: integrating machine translation effectively and imperceptibly. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2017. An exploration of neural sequence-to-sequence architectures for automatic post-editing. *arXiv preprint arXiv:1706.04138*.

Rebecca Knowles and Philipp Koehn. 2016. Neural interactive translation prediction. In *AMTA (1)*, pages 107–120.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistic*, pages 127–133, Edmondton, Canada.

Huayang Li, Lemao Liu, Guoping Huang, and Shuming Shi. 2021. Gwlan: General word-level autocompletion for computer-aided translation. *arXiv preprint arXiv:2105.14913*.

António V Lopes, M Amin Farajian, Gonçalo M Correia, Jonay Trénous, and André FT Martins. 2019. Unbabel's submission to the wmt2019 ape shared task: Bert-based encoder-decoder for automatic post-editing. *arXiv preprint arXiv:1905.13068*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, page 1412–1421.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. *Proceedings of NAACL-HLT 2018*, page 1314–1324.

Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón, and Sergio Ortiz-Rojas. 2020. Bifixer and bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298, Lisboa, Portugal. European Association for Machine Translation.

Nipun Sadvilkar and Mark Neumann. 2020. PySBD: Pragmatic sentence boundary disambiguation. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 110–114, Online. Association for Computational Linguistics.

Sebastin Santy, Sandipan Dandapat, Monojit Choudhury, and Kalika Bali. 2019. Inmt: Interactive neural machine translation prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 103–108.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, page 1715–1725.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Tianxiang Zhao, Lemao Liu, Guoping Huang, Huayang Li, Yingling Liu, Liu GuiQuan, and Shuming Shi. 2020. Balancing quality and human involvement: An effective approach to interactive neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9660–9667.

Translation Word-Level Auto-Completion: What can we achieve out of the box?

Yasmin Moslem
ADAPT Centre
School of Computing
Dublin City University
Dublin, Ireland
yasmin.moslem@adaptcentre.ie

Rejwanul Haque
School of Computing
National College of Ireland
Mayor Street, IFSC
Dublin, Ireland
rejwanul.haque@ncirl.ie

Andy Way
ADAPT Centre
School of Computing
Dublin City University
Dublin, Ireland
andy.way@adaptcentre.ie

Abstract

Research on Machine Translation (MT) has achieved important breakthroughs in several areas. While there is much more to be done in order to build on this success, we believe that the language industry needs better ways to take full advantage of current achievements. Due to a combination of factors, including time, resources, and skills, businesses tend to apply pragmatism into their AI workflows. Hence, they concentrate more on outcomes, e.g. delivery, shipping, releases, and features, and adopt high-level working production solutions, where possible. Among the features thought to be helpful for translators are sentence-level and word-level translation auto-suggestion and auto-completion. Suggesting alternatives can inspire translators and limit their need to refer to external resources, which hopefully boosts their productivity. This work describes our submissions to WMT's shared task on word-level auto-completion, for the Chinese-to-English, English-to-Chinese, German-to-English, and English-to-German language directions. We investigate the possibility of using pre-trained models and out-of-the-box features from available libraries. We employ random sampling to generate diverse alternatives, which reveals good results. Furthermore, we introduce our open-source API, based on CTranslate2, to serve translations, auto-suggestions, and auto-completions.

1 Introduction

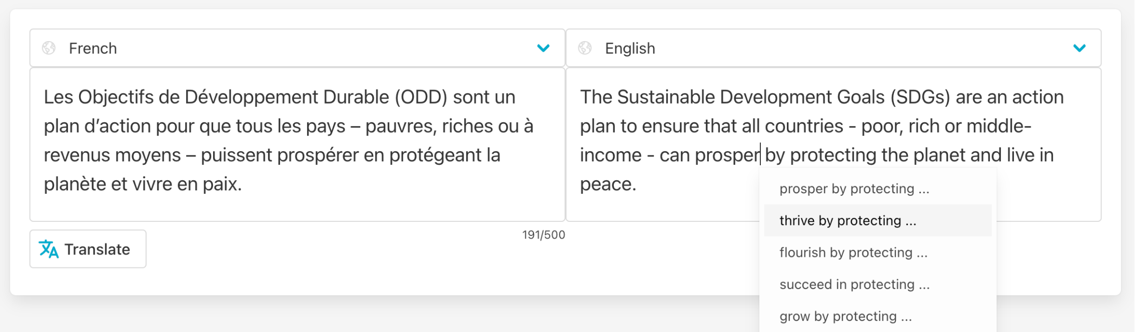
Translation auto-suggestion and auto-completion are among the important features that can help translators better utilize Machine Translation (MT) systems. In a Computer-Aided Translation (CAT) environment, a translator can make use of the MT word auto-suggestion feature as follows:

- typing a few words, or clicking a word in a proposed MT translation, a list of suggestions is displayed, as illustrated by Figure 1.
- selecting one of the word suggestions from the list, the rest of the translation is modified accordingly.

The WMT's Word-Level AutoCompletion (WLAC) shared task addresses a more specific scenario, where the user types a few characters, and the system predicts and auto-completes the correct word, given the current context. The WLAC task even suggests that the context might be partial, and it can consist of preceding and/or following words. Given a source sequence x, typed character sequence s and a context c, WLAC aims to predict a target word w which is to be placed in the middle between the left context $\mathrm{c_l}$ and right context $\mathrm{c_r}$ to constitute a partial translation. Note that the last word of $\mathrm{c_l}$, the auto-completed word w, and the first word of $\mathrm{c_r}$ are not necessary consecutive.

Previous work proposed diverse approaches, mostly to translation sentence-level auto-suggestion and auto-completion. In their work, Li et al. (2021) proposed an approach to tackle the word-level auto-completion task. Given a tuple (x, c, s), the system decomposes the word autocompletion process into two parts: 1) model the distribution of the target word w based on the source sequence x and the translation context c; and 2) find the most possible word w based on the distribution and human typed sequence s. Hence, they first use a single placeholder [MASK] to represent the unknown target word w, and use the representation of [MASK] learned from the word prediction model, based on BERT (Devlin et al., 2019), to predict it. Then, the predicted distribution of the masked token is used over the vocabulary to filter out invalid words, namely those that do not start with the human typed sequence s.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1176 - 1181
December 7-8, 2022 ©2022 Association for Computational Linguistics



Les Objectifs de Développement Durable (ODD) sont un plan d'action pour que tous les pays – pauvres, riches ou à revenus moyens – puissent prospérer en protégeant la planète et vivre en paix.

The Sustainable Development Goals (SDGs) are an action plan to ensure that all countries - poor, rich or middle-income - can prosper by protecting the planet and live in peace.

French
English
191/500
Translate
prosper by protecting ...
thrive by protecting ...
flourish by protecting ...
succeed in protecting ...
grow by protecting ...

Figure 1: Auto-Suggest: Word Suggestions List[1]

Finally, they return the token with the highest probability over the new distribution.

Researchers in other natural language processing areas such as language modelling offered approaches to improve predictions of decoder-only autoregressive models, trained to predict the next word given the previous context. Among these approaches are top-K sampling and top-p (nucleus) sampling (Fan et al., 2018; Holtzman et al., 2018; Radford et al., 2019; Holtzman et al., 2020). Since neural machine translation inference depends on a decoder model, such approaches from language modelling can be employed. In particular, we investigate the use of top-K sampling during decoding to generate better word-level auto-completions.

2 User Survey

Previous work reported that a user can save over 60% of the keystrokes needed to produce a translation in a word completion scenario (Langlais et al., 2000). Other researchers noted that post-editing is faster than MT auto-completion (Koehn, 2009) while MT auto-completion can yield higher quality translation when the baseline MT quality is high (Green et al., 2014).

In a user survey we designed and distributed via social media networks, we asked participants whether they thought an MT word-level auto-suggestions feature could be helpful, and provided a simple definition and an illustrative image. If their answer was "yes", the respondent was asked to specify a reason. By the time of writing this paper, we received 41 responses to our survey. While we do not believe this survey is enough to justify introducing an auto-suggestions feature into every MT system, it can be an indicator as to why some users think such a feature could be helpful. To answer the question, "Which of the following best describes you?" 46.3% (19) of the respondents chose "Translator/Linguist", 31.7% (13) selected "NLP Engineer/Researcher", and the rest 22% (9) were other "MT Users", not included in the two aforementioned categories.

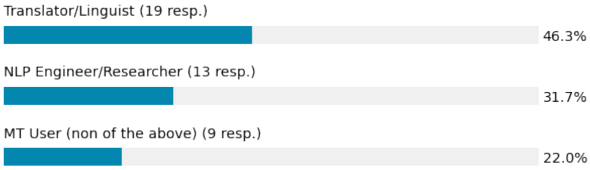


Figure 2: MT user categories

Among the respondents to the survey, 90.2% (37) answered "Yes" to the question "In general, do you believe that a word-level auto-suggestions feature is helpful?" Figure 3 shows the breakdown of answers to the question, "Why do you believe that a word-level auto-suggestions feature can be helpful?" taking into consideration those who answered "No" to the previous question.

Out of the 37 persons who believed a word-level auto-suggestions feature can be helpful, 40.5% (15) of the respondents specified that it can give them some inspiration. This answer is specifically interesting as it is not constrained by time-saving benefits; hence, it focusses more on effectiveness rather than efficiency. The respondent that answered with "Other" mentioned that it allows them to look for alternative senses or phrasings, especially when they suspect the initial translation is bad, and referred to this as "human in the loop".

Respondents were allowed to give extra comments; among the notable comments were:

[1] The image is from our demo at: https://www.machinetranslation.io/

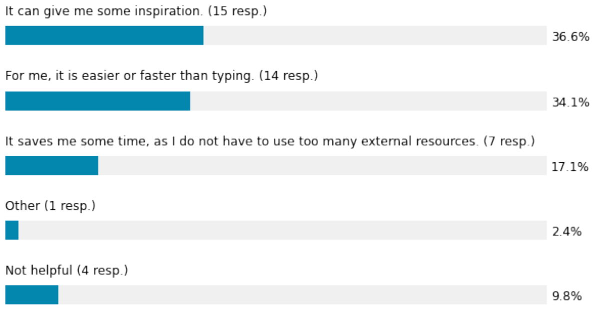


Figure 3: How translators and other MT users perceive word-level auto-suggestions

- *I think word-level suggestions can be a useful feature, particularly when the target language can have several translations of a single source word.*
- *Word-level suggestions can be helpful, but sometimes you end up spending a lot of time figuring out if the MT suggestion is a valid translation in that context. So, I'm not really sure yet how I feel about it.*
- *It's useful, as long as it's seen as a suggestion, and not inserted in the target where the translator is typing.*

Among the respondents who chose "For me, it is easier or faster than typing", comments included:

- *Though most of the time; the suggestions are lousy.*
- *I don't think it gives me inspiration as I mostly need it for structures, not single words.*
- *Auto-suggestion does not have to come from machine translation. History is much more useful.*

The last comment above might be referring to the fact that in some CAT tools, auto-suggestions can also include glossary terms, and translation memory sub-segments, which encourages further research efforts to investigate methods to enhance leveraging and interaction between various translation resources in human-in-the-loop environments.

We hope this survey will inspire future user studies to look deeper into how diverse users of MT and CAT tools prefer to utilize certain features, such as auto-suggestions, and the value they seek. More aspects should be taken into consideration such as language pairs, translation workflows, and user interfaces. This can help improve these features to better support linguists and other MT users and boost their productivity as well as translation quality.

3 Experimental Setup

Models We use OPUS pre-trained models[2] based on the Transformer architecture (Vaswani et al., 2017) for the Chinese-to-English, English-to-Chinese, German-to-English, and English-to-German language directions.

Tokenizers OPUS models depend on SentencePiece[3] (Kudo and Richardson, 2018) for tokenization. Hence, we use their provided subword models during our pre-processing and post-processing processes. As OPUS's English-to-Chinese model requires defining the target dialect using a pre-specified token, we prepend [">>cmn_Hans<<"] to the list of tokens generated by SentencePiece. For word-level tokenization, we use NLTK for English and German, and Jieba[4] for Chinese. This list of words can be used later to find the word that starts with the typed sequence.

Inference Engine We employ CTranslate2 (Klein et al., 2020) for sentence-level MT, as well as for translation auto-suggestions. To this end, we first convert OPUS models into the CTranslate2 format. After that, we utilize a number of CTranslate2 decoding features, including "alternatives at a position" and "auto-completion".[5] The translation options $return\_alternatives$ and $num\_hypotheses$ are essential for all our experiments; the former should be set to $True$ while the latter determines the number of returned alternatives. These decoding options can be used with regular beam search, prefix-constrained decoding, and/or random sampling. If the decoding option $return\_alternatives$ is used along with $target\_prefix$, the provided target left context is fed into the decoder in the teacher forcing mode,[6] then the engine expands the next N most likely words, and continues (auto-completes) the decoding for these N hypotheses independently. The shared task investigates four context cases:

[2] https://github.com/Helsinki-NLP/Tatoeba-Challenge

[3] https://github.com/google/sentencepiece

[4] https://github.com/fxsjy/jieba

[5] https://github.com/OpenNMT/CTranslate2/blob/master/docs/decoding.md

[6] In *teacher forcing* (Williams and Zipser, 1989), ground truth previous tokens are fed into the decoder, instead of the predicted tokens y_{i-1} as suggested by Bahdanau et al. (2015)

(a) empty context, (b) right context only, (c) left context only, and (d) both the right and left contexts are provided. Hence, for all cases we returned multiple alternative translations, while for (c) and (d) we also returned another set of alternative autocompletions using the left context as a target prefix. In this sense, it is worth noting that we make use only of the left context, when available, and we do not use the right context at all, which we might investigate further in the future. To enhance diversity of translations, especially for (a) and (b), we applied random sampling with the CTranslate2's decoding option $sampling\_topk$, with various sampling temperatures. Our experiments are further elaborated in Section 4 and Section 5.

Pinyin The official Romanization system for Standard Mandarin Chinese is called Pinyin. Since the task organizers used the pypinyin library[7] to prepare the test files, we did too. OPUS English-to-Chinese models accept Chinese input, so we had to use the library to convert between the two writing systems. Since the conversion from Chinese characters to Pinyin is a lossy process and cannot be perfectly converted back, we keep a list of Chinese words resulted from tokenization with Jieba to be able to map Pinyin tokens to Chinese tokens later.

4 Method

We experimented with both beam search alternatives and random sampling, and found that the latter achieves better results. This could be due to the fact that alternatives generated from each beam are usually very similar, and lower beam values tend to generate translations of lower quality. This section elaborates on the actual methods we used for our submissions, while more details about initial experiments that led us to these decisions are explained in Section 5.

Random sampling is a decoding mode that randomly samples tokens from the model output distribution. In our experiments, we restrict the sampling to the top-10 candidates at each time-step. To obtain diverse generations from the MT model, we rely on randomness in the decoding method, in particular through top-K sampling that samples the next word from the top-K most probable choices (Fan et al., 2018; Holtzman et al., 2018; Radford et al., 2019), instead of aiming to decode text that maximizes likelihood.

For each translation, we use the CTranslate2 option *return_alternatives* to return 10 sequences, with 10 top-K sampling. If the entry has a left context starting with a capital letter, we use the prefix to constrain the decoding. In CTranslate2, combining *target_prefix* with the *return_alternatives* flag returns alternative sequences just after the prefix. We compose a list of alternatives with and without the prefix, and try to find the word starting with the typed sequence.[8] If the word is not found, we repeat the same process for up to five runs. In each new run, random sampling can generate a new set of alternatives. Our experiments show that returning 20 sequences with 20 top-K sampling could lead to more correctly predicted words (cf. Table 2); however, we had to consider the trade-off between quality and efficiency.[9]

Furthermore, we investigate increasing the randomness of the generation by using a value for sampling temperature between 1.0 and 1.3. For each run, a random value is generated in this range. The default sampling temperature in CTranslate2 is 1, which achieved relatively better results, as demonstrated in Table 1.

Language	Settings	Accuracy	Human
de-en	ST=1.0	0.614441141	0.885
	ST=1.3	0.609237735	0.8875
en-de	ST=1.0	0.589418807	0.6725
	ST=1.3	0.584939177	0.655
zh-en	ST=1.0 + detok	0.504113456	0.8675
	ST=1.3 + detok	0.502598878	0.8675
	ST=1.0	0.493476989	0.86
	ST=1.3	0.490619944	0.87
en-zh	ST=1.0	0.319424091	0.5775
	ST=1.3	0.319350821	0.5725

Table 1: Evaluation results on the test datasets. Automatic evaluation uses the "Accuracy" metric. "Human" refers to human evaluation. Results obtained from sampling temperature (ST) 1.0 are slightly better than those with the value 1.3. When the source is Chinese, detokenization (detok) resulted in slightly better scores.

[7] https://github.com/mozillazg/python-pinyin

[8] In a prefix-free target sequence, if multiple words start with the typed sequence, we return the first word. In practice, users could be prompted to choose from potential options.

[9] Our scripts are available at: https://github.com/ymoslem/WLAC

5 Other Experiments

This section elaborates on some initial experiments we conducted to decide what approach to use. The final approach we actually used in our submissions is explained in Section 4.

We used 10,000 entries of a Chinese-to-English golden sample provided by the organizers to evaluate various experiments. For sentence translation, when there is no left context, we experimented with the following values:

- beam size 1, 5, and 10, without sampling
- beam size 1, with random sampling top-K 10, 20, and 50

Table 2 shows the results for these experiments, and demonstrates that random sampling achieves the best overall accuracy. Random sampling with beam size 1 reveals better results than mere beam size 1 and even beam sizes 5 and 10 without random sampling. Multiple runs of random sampling can result in more correctly predicted words.

Beam Size	Sampling Top-K	Hypotheses	Accuracy	Runs
1	N/A	10	0.6519	1
5	N/A	10	0.6588	1
10	N/A	10	0.6573	1
1	10	10	0.6918	1
1	20	10	0.6907	1
1	**20**	**20**	**0.7108**	**1**
1	50	10	0.6853	1
5	N/A	10	0.6588	5
1	10	10	0.7165	5
1	**20**	**20**	**0.7310**	**5**

Table 2: Results for the Chinese-to-English golden sample dataset (10,000 entries). Random sampling outperforms even higher beam sizes.

6 API

Our API project[10] offers an easy way to integrate translation, auto-suggestion, and auto-completion features into translation environments. We chose FastAPI[11] for its high performance that beats many other Python web frameworks[12] in addition to its easy integration with OpenAPI (Swagger) documentation.

6.1 API Endpoints

The API consists of a number of endpoints, receiving requests and sending the relevant responses in the JSON format. Each of the MT features has its endpoint.

6.1.1 Translation Endpoint

The API handles a POST request (e.g. received from a CAT environment), including:

- $sentences$: list of the source sentences to be translated.
- $source\_language$: in a format like “fr” for French, and the default is “auto” to run language auto-detection.
- $target\_language$: in a format like “en” for English.

The API response is a list of strings for the MT translations in a JSON format.

6.1.2 Auto-Suggestions Endpoint

When the user clicks on one word of the MT translation, the CAT environment sends a request to the API including:

- $sentence$: sentence to be translated.
- $prefix$: words to start the translation with.
- $source\_language$: in a format like “fr” for French, and the default is “auto” to run language auto-detection.
- $target\_language$: in a format like “en” for English.

The API response is a list of the MT word suggestions/alternatives for the current word, and the translation auto-completions if the user selects a specific suggestion.

6.2 JSON Response Examples

This is an example of a response to the translation request referred to in Section 6.1.1.

```
{ ‘id’: 10550004
  ‘source_lang’: "fr",
  ‘target_lang’: "en",
  ‘translations’: [
    ‘The COVID−19 crisis has deepened already
         existing inequalities.’
  ]
}
```

This is an example of a response to the auto-suggestions request referred to in Section 6.1.2.

[10] https://github.com/ymoslem/SnowballMT

[11] https://github.com/tiangolo/fastapi

[12] https://www.techempower.com/benchmarks/#section=data-r20&hw=ph&test=query&l=zijzen-sf

```
{
  ‘id’: 10550005,
  ‘source_lang’: "fr",
  ‘target_lang’: "en",
  ‘result’: {
    ‘translations’: [
      {
        ‘suggestion’: ‘crisis’,
        ‘compelection’: ‘of COVID−19 has deepened
            already existing inequalities.’
      },
      {
        ‘suggestion’: ‘COVID−19’,
        ‘compelection’: ‘crisis has deepened already
             existing inequalities.’
      },
      {
        ‘suggestion’: ‘impact’,
        ‘compelection’: ‘of COVID−19 crisis has
            deepened already existing inequalities
            .’
      }
    ]
  }
}
```

Acknowledgements

This work is supported by the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224, the ADAPT Centre for Digital Content Technology which is funded under the Science Foundation Ireland (SFI) Research Centres Programme (Grant No. 13/RC/2106) and is co-funded under the European Regional Development Fund, and Microsoft Research.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, USA.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Spence Green, Sida I Wang, Jason Chuang, Jeffrey Heer, Sebastian Schuster, and Christopher D Manning. 2014. Human Effort and Machine Learnability in Computer Aided Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1225–1236, Doha, Qatar. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, Virtual.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to Write with Cooperative Discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1638–1649, Melbourne, Australia. Association for Computational Linguistics.

Guillaume Klein, François Hernandez, Vincent Nguyen, and Jean Senellart. 2020. The OpenNMT Neural Machine Translation Toolkit: 2020 Edition. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 102–109, Virtual. Association for Machine Translation in the Americas.

Philipp Koehn. 2009. A process study of computer-aided translation. *Machine Translation*, 23(4):241–263.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Philippe Langlais, George Foster, and Guy Lapalme. 2000. TransType: a Computer-Aided Translation Typing System. In *ANLP-NAACL 2000 Workshop: Embedded Machine Translation Systems*.

Huayang Li, Lemao Liu, Guoping Huang, and Shuming Shi. 2021. GWLAN: General Word-Level AutocompletioN for Computer-Aided Translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4792–4802, Online. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, D Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. Technical report, OpenAi.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems (NIPS 2017)*, volume 30. Curran Associates, Inc.

Ronald J Williams and David Zipser. 1989. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Comput.*, 1(2):270–280.

PRHLT's Submission to WLAC 2022

Ángel Navarro and **Miguel Domingo** and **Francisco Casacuberta**
PRHLT Research Center
Universitat Politècnica de València
{annamar8,midobal,fcn}@prhlt.upv.es

Abstract

This paper describes our submission to the Word-Level AutoCompletion Shared Task of the WMT 2022. We participate in the pair of languages, English–German, in both ways. We propose a segment-based interactive machine translation approach whose central core is a machine translation (MT) model that predicts the complete translation from the context provided by the task and picks the word we were trying to autocomplete from there. We show with this approach that it is possible to use the MT models in the autocompletion task by performing minor changes at the decoding step and obtaining good accuracy.

1 Introduction

Machine translation (MT) has significantly improved in recent years with the emergence of neural machine translation (NMT), but it still cannot assure high-quality translations for all tasks (Toral, 2020). For those scenarios with rigorous translation quality requirements, it is critical for professional translators to manually validate the translations generated by the NMT system. The computer-aided translation (CAT) tools show up to improve the validation and editing process carried out by translators. Researchers approached CAT tools from many directions with the aim of reducing the human effort of correcting the automatic translations. Among CAT tools such as translation memory (Zetzche, 2007), augmented translation (Lommel, 2018) and terminology management (Verplaetse and Lambrechts, 2019); we can find autocompletion tools, which help professional translators by providing new partial translations according to the validated parts they have supplied to the system.

Word level autocompletion (WLAC) (Li et al., 2021) is a new shared task introduced in WMT22. Its aim is to complete a target word given a source sentence, a sequence of characters typed by the human translator and a translation context. Four types of context are possible:

Zero-contex: no context is given.

Suffix: a sequence of translated words located after the word to autocomplete.

Prefix: a sequence of translated words located prior to the word to autocomplete.

Bi-contex: A combination of the *suffix* and the *prefix* type. That is, there is a sequence of translated words located after the word to autocomplete, and a sequence of translated words located prior to the word to autocomplete.

Note that, in all cases, the word to autocomplete is not necessarily consecutive to these contexts.

We have experimented with a similar CAT tool from the interactive machine translation (IMT) framework. In this field of research, the translation is generated in a collaborative process between the human translator and the MT model. Among the different approaches, the segment-based IMT (Domingo et al., 2017; Peris et al., 2017) protocol presents certain similarities with WLAC: at each step, the user validates sequences of translated words—the context—and makes a correction—the word to autocomplete.

Therefore, in this work we have approached WLAC as a simplification of segment-based IMT, using the context as the validated segments and the typed characters as the word correction; and limiting the process to the first iteration. This has allowed us to tackle WLAC by training a conventional NMT model and adapting it at the decoding step.

2 Segment-based interactive machine translation

Segment-based IMT establishes a framework in which a human translator works together with

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1182 - 1186
December 7-8, 2022 ©2022 Association for Computational Linguistics

SOURCE (x):		Una versión traducida de un texto.
REFERENCE (y):		A translated version of a text.
ITER-0	($\tilde{\mathbf{f}}$)	( )
	($\hat{y}$)	A written version of a story.
ITER-1	($\tilde{\mathbf{f}}$)	(A) (version of a)
	(s)	t
	($\hat{y}$)	A translated version of a document.
ITER-2	($\tilde{\mathbf{f}}$)	(A translated version of a)
	(s)	t
	($\hat{y}$)	A translated version of a text.
FINAL	($\hat{y} \equiv y$)	A translated version of a text.

(a) Segment-based IMT session.

c_l	s	c_r
A	t	version of a
A	translated	version of a

(b) WLAC session.

Figure 1: Examples of a segment-based IMT session to translate a sentence from Spanish to English; and a WLAC session for predicting a word for a source sentence, a translation context, and a human-typed character sequence.

the MT system to produce the final translation. This collaboration starts with the system proposing an initial translation hypothesis y_1^I of length I. Then, the user reviews this hypothesis and validates those sequence of words which they consider to be correct ($\tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_N$; where N is the number of non-overlapping validated segments). After that, they are able to merge two consecutive segments $\tilde{\mathbf{f}}_i$, $\tilde{\mathbf{f}}_{i+1}$ into a new one. Finally, they correct a word—which introduces a new one-word validated segment, $\tilde{\mathbf{f}}_i$, which is inserted in $\tilde{\mathbf{f}}_1^N$. This correction can also consist in a partially typed word $\tilde{\mathbf{f}}'_i$, in which case the system would complete it as part of its prediction.

The system's reacts to this user feedback by generating a sequence of new translation segments $\widehat{\mathbf{g}}_1^N = \widehat{\mathbf{g}}_1, \ldots, \widehat{\mathbf{g}}_N$; where each $\widehat{\mathbf{g}}_n$ is a subsequence of words in the target language. This sequence complements the user's feedback to conform the new hypothesis:

$$\begin{cases} \hat{y}_1^I = \tilde{\mathbf{f}}_1, \widehat{\mathbf{g}}_1, \ldots, \tilde{\mathbf{f}}'_i \widehat{\mathbf{g}}_i, \ldots, \tilde{\mathbf{f}}_N, \widehat{\mathbf{g}}_N \text{ if } \tilde{\mathbf{f}}'_i \in \tilde{\mathbf{f}}_1^N \\ \hat{y}_1^I = \tilde{\mathbf{f}}_1, \widehat{\mathbf{g}}_1, \ldots, \tilde{\mathbf{f}}_N, \widehat{\mathbf{g}}_N \text{ otherwise} \end{cases} \tag{1}$$

The word probability expression for the words belonging to a validated segment $\tilde{\mathbf{f}}_n$ was formalized by Peris et al. (2017) as:

$$p(y_{i_n+i'} \mid y_1^{i_n+i'-1}, x_1^J, f_1^N; \Theta) = \mathbf{y}_{i_n+i'}^\top \mathbf{p}_{i_n+i'}, \quad 1 \le i' \le \hat{l}_n \tag{2}$$

where l_n is the size of the non-validated segment generated by the system, which is computed as follows:

$$\hat{l}_n = \underset{0 \le l_n \le L}{\arg\max} \frac{1}{l_N + 1} \sum_{i'=i_n+1}^{i_n+l_n+1} \log p(y_{i'} \mid y_1^{i'-1}, x_1^J; \Theta) \tag{3}$$

3 Approach

Given a source sentence x_1^J, a sequence of typed characters $s_1^K = s_1, \ldots, s_K$ and a context $\mathbf{c} = \{\mathbf{c}_\mathrm{l}, \mathbf{c}_\mathrm{r}\}$, where $\mathbf{c}_\mathrm{l} = c_{\mathrm{l}1}, \ldots, c_{\mathrm{l}S}$ and $\mathbf{c}_\mathrm{r} = c_{\mathrm{r}1}, \ldots, c_{\mathrm{r}R}$; WLAC aims to autocomplete s_1^K to conform the word $w_1^W = s_1, \ldots, s_K, w_{K+1}, \ldots, w_W$. If we consider the context as the sequence of segments validated by the user ($\tilde{\mathbf{f}}_1^N = \mathbf{c}_\mathrm{l}, \mathbf{c}_\mathrm{r}$) and the sequence s_1^K as the partially-typed word correction (which would be inserted in $\tilde{\mathbf{f}}_1^N$ as a new one-word validated segment; leading to $\tilde{\mathbf{f}}_1^N = \mathbf{c}_\mathrm{l}, s_1^K, \mathbf{c}_\mathrm{r}$), we can view WLAC as a simplification of segment-based IMT. With that in mind, we can rewrite Eq. (1) as:

$$\hat{y}_1^I = \mathbf{c}_\mathrm{l}, \widehat{\mathbf{g}}_1, s_1^K \widehat{\mathbf{g}}_2, \mathbf{c}_\mathrm{r}, \widehat{\mathbf{g}}_3 \tag{4}$$

which, knowing that the prediction of the partially-typed correction corresponds to the first word of $\widehat{\mathbf{g}}_2$, can be rewritten as:

$$\hat{y}_1^I = \mathbf{c}_\mathrm{l}, \widehat{\mathbf{g}}_1, s_1^K w_{K+1}^W, \widehat{\mathbf{g}}'_2, \mathbf{c}_\mathrm{r}, \widehat{\mathbf{g}}_3 \tag{5}$$

Therefore, we can obtain the autocompleted word ($w_1^W = s_1^K w_{K+1}^W$) by performing a single step of the segment-based IMT protocol, discarding the rest of the translation prediction.

Figure 1 illustrates an example of segment-based IMT compared to the WLAC task for the same case.

In the segment-based IMT (Fig. 1a) example, at iteration 0, the system generates an incorrect first hypothesis. The, at iteration 1, the user validates a sequence of segments and types the first character of the word 'translated' to help the system fulfill the sequence of words between the first two segments. After that, the system generates a new translation with all the validated segments and the human-typed character sequence. The system repeats this process at the second iteration, ending with a correct translation. The WLAC (Fig. 1b) example simplifies the case that happens at iteration 1. Although we have the same source sentence, validated segments (left and right context) and human-typed character sequence, in this case, the system only has to find one word between the two segments instead of generating the whole sentence.

4 Experimental setup

In this section, we present the details of our experimental session.

4.1 Evaluation

The WLAC 22 shared task selected accuracy as the automatic metric with which to report the evaluation of the different systems[1]. This metric is computed as the total number of correctly predicted words normalized by the total number of words to complete:

$$\text{Acc} = N_{\text{match}}/N_{\text{all}} \tag{6}$$

where N_{match} is the number of predicted words that are identical to the human desired word, and N_{all} is the total number of testing words.

4.2 Corpora

We conducted our experiments using the English–German corpus provided by the organizers, which is a version of the WMT14's dataset, preprocessed by Stanford NLP Group. We saved 2000 sentences to use as validation, which we processed with the provided script[2] in order to create the simulated data. Table 1 presents the data statistics.

Table 1: Statistics of the WLAC 2022 corpus. *Avg.* stands for average, *Run.* for running, K for thousands and M for millions.

Partition	**Characteristic**	**De**	**En**
Training	Sentences	$4M$	
	Avg. Length	25	26
	Run. Words	$110M$	$116M$
	Vocabulary	$1.6M$	$800K$
Validation	Sentences	2000	
	Avg. Length	27	27
	Run. Words	$53K$	$53K$

4.3 Systems

Our MT systems were trained using *OpenNMT-py* (Klein et al., 2017). We made use of two different network architectures: recurrent neural network (RNN) (Hochreiter and Schmidhuber, 1997) and Transformer (Vaswani et al., 2017).

The RNN model uses an encoder–decoder architecture with long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) cells. We set the size of the encoder, decoder and word embedding layers to 512. The encoder and decoder models use a single hidden layer of the same size. We used Adam (Kingma and Ba, 2014) as the learning algorithm, with a learning rate of 0.0002 with a batch size of 10.

The Transformer model uses a word embedding size of 512. The hidden and output layers were set to 2048 and 512, respectively. Each multi-head attention layer has eight heads, and we stacked six encoder and decoder layers. We used Adam as the learning algorithm, with a learning rate of 2.0, b_1 of 0.9 and b_2 of 0.998. We set the batch size to 4096 tokens.

Additionally, we made use of the byte pair encoding (BPE) (Sennrich et al., 2016) algorithm, which was jointly trained on both languages of the dataset, applying a maximum number of 10.000 merges.

Finally, we used our own implementation (based on *OpenNMT-py*) of segment-based IMT, which we adapted for WLAC. This implementation is openly available[3] for the benefit of the community.

5 Results

In this section we present our experimental results. We trained four different models, alternating between the RNN and Transformer architectures and the use of the BPE algorithm on the En-

[1] A human evaluation was also performed.

[2] https://github.com/lemaoliu/WLAC/raw/main/scripts/generate_samples.py.

[3] https://github.com/PRHLT/OpenNMT-py/tree/word-level_autocompletion.

Table 2: Experimental results, measured in terms of accuracy. Test values are taken from the official evaluation. Best results from the validation set are denoted in **bold**.

Partition	Approach	De–En	En–De
Validation	RNN	0.568	**0.535**
	RNN + BPE	0.554	0.498
	Transformer	0.563	0.524
	Transformer + BPE	**0.586**	**0.534**
Test	Transformer + BPE	0.390	0.340

glish–German language pairs.

Prior to submitting our systems, we used the synthetic validation dataset created from the provided data (see Section 4.2). As reflected in Table 2, all approaches yielded similar results. They correctly completed the word the user was trying to type around 60% of the time. Since the *Transformer + BPE* combination yielded a two points improvement for De–En, and also achieved—together with the *RNN* approach—the best results for En–De, we selected this model for our submission.

Table 2 also contains the official accuracy scores published by the organizers. For the blind test, our system's performance dropped near a 20%. While we are waiting for the publication of the findings to have a better understanding of the cause of this drop, we suspect that it is related with the test set being from a different domain than the training data, which would have a considerable impact in our MT model.

All in all, these results show that the segment-based IMT methodology is a promising approach to adapting an MT model to the WLAC task. Moreover, due to the shared task constrains, we trained our systems using only the data provided by the organizers. However, one of the benefits of our approach is that any MT system can be easily adapted to be used for WLAC.

6 Conclusions

In this work, we have presented our submission to WLAC shared task from WMT22. Our approach consisted in adapting the segment-based IMT methodology to the WLAC task, which allows us to use a conventional NMT model to tackle this task by simply adapting it at the decoding step. We tested some of the most used NMT architectures, achieving very encouraging results.

As a future work, we would like to test our approach using a more robust NMT system, adapted to the domain of the task to perform—instead of training an ad hoc system, as we did in this work due to the task restrictions.

Acknowledgements

The research leading to these results has received funding from *Generalitat Valenciana* under project *PROMETEO/2019/121*.

References

Miguel Domingo, Álvaro Peris, and Francisco Casacuberta. 2017. Segment-based interactive-predictive machine translation. *Machine Translation*, 31:1–23.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proceedings of the Association for Computational Linguistics: System Demonstration*, pages 67–72.

Huayang Li, Limao Liu, Guoping Huang, and Shuming Shi. 2021. GWLAN: General Word-Level AutocompletioN for computer-aided translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 4792–4802.

Arle Lommel. 2018. Augmented translation: A new approach to combining human and machine capabilities. In *Proceedings of the Conference of the Association for Machine Translation in the Americas. Volume 2: User Track*, pages 5–12.

Álvaro Peris, Miguel Domingo, and Francisco Casacuberta. 2017. Interactive neural machine translation. *Computer Speech & Language*, 45:201–220.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.

Antonio Toral. 2020. Reassessing claims of human parity and super-human performance in machine translation at WMT 2019. *arXiv preprint arXiv:2005.05738*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Heidi Verplaetse and An Lambrechts. 2019. Surveying the use of CAT tools, terminology management systems and corpora among professional translators: general state of the art and adoption of corpus support by translator profile. *Parallèles*, 31(2):3–31.

Jost Zetzche. 2007. Translation memory: state of the technology. *Multilingual*, 18:34–38.

IIGROUP Submissions for WMT22 Word-Level AutoCompletion Task

Cheng Yang   Siheng Li   Chufan Shi   Yujiu Yang
Tsinghua Shenzhen International Graduate School, Tsinghua University
{yangc21,lisiheng21,scf22}@mails.tsinghua.edu.cn
yang.yujiu@sz.tsinghua.edu.cn

Abstract

This paper presents IIGroup's submission to the WMT22 Word-Level AutoCompletion(WLAC) Shared Task in four language directions. We propose to use a *Generate-then-Rerank* framework to solve this task. More specifically, the generator is used to generate candidate words and recall as many positive candidates as possible. To facilitate the training process of the generator, we propose a span-level mask prediction task. Once we get the candidate words, we take the top-K candidates and feed them into the reranker. The reranker is used to select the most confident candidate. The experimental results in four language directions demonstrate the effectiveness of our systems. Our systems achieve competitive performance ranking 1st in English to Chinese subtask and 2nd in Chinese to English subtask.

1 Introduction

Recent advances in neural machine translation (Bahdanau et al., 2015; Vaswani et al., 2017) allow us to generate high-quality translation results. However, as it's pointed out by Li et al. (2021) that in some scenarios(e.g., legal instruments), the results of machine translation can't directly replace human translations due to their imperfections(e.g., terminology translation error). Therefore, more and more researchers pay attention to *Computer-aided translation*(CAT)(Barrachina et al., 2009; Santy et al., 2019; Huang et al., 2021; Xiao et al., 2022), which focuses on leveraging the advantages of NMT systems to increase the effectiveness and efficiency of the human translation process.

To further promote the development of CAT, WMT22 proposes a novel task —— Word-Level AutoCompletion(WLAC). In the Word-Level AutoCompletion task, given a source sentence $\boldsymbol{x}$, target context and human-typed characters $\boldsymbol{t}$, an ideal system is expected to be able to predict the target word w that should be placed in the target context.

We participate in the WMT22 shared Word-Level AutoCompletion task in four language directions: Chinese ⇒ English, English ⇒ Chinese, German ⇒ English and English ⇒ German and submit a system for each language direction.

We develop a *Generate-then-Rerank* framework-based system for each language direction. Based on the vanilla Transformer architecture, we adopt a bidirectional decoder, which can predict the current target word by paying attention to the source sentence and both the left-side and right-side target context.

The paper is organized as follows, section 2 gives the overview of the data used in the shared task and preprocessing operations for the data, while section 3 describes our training techniques, including model architecture, span-level mask prediction, etc. Section 4 presents our experimental results. Finally, our conclusions are summarized in Section 5.

2 Data

In this section, we first introduce the datasets used to train our systems, then we introduce how to prepare the simulated training data for the WLAC shard task and describe the vocabulary for each language direction.

2.1 Datasets

As the WLAC shared task is a data-constrained task, we can only use the parallel corpora provided by the WLAC organizers for all four language directions. Specifically, we use UN Parallel Corpus V1.0[1] (WMT 2017) for Chinese ⇒ English and English ⇒ Chinese. For German ⇒ English and English ⇒ German, we use the WMT 14 dataset pre-processed by Stanford NLP Group[2]. Details of the training resources provided are shown in Table 1.

[1] https://conferences.unite.un.org/uncorpus

[2] https://nlp.stanford.edu/projects/nmt

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1187 - 1191
December 7-8, 2022 ©2022 Association for Computational Linguistics

	Zh-En	De-En
Train Set	10M	4.5M
Validation Set	3k	3k

Table 1: The detailed statistics of training and validation data used in our system.

	Zh⇒En	En⇒Zh	De⇒En	En⇒De
source	60k	50k	50k	50k
target	50k	60k	50k	50k

Table 2: The vocabulary size of different language directions.

2.2 Simulated Training Data

Since the WLAC shared task only provides raw parallel corpora and does not provide supervised data, which complies with the WLAC shared task setting, we need to automatically construct supervised data from the raw parallel corpora.

Specifically, given a raw parallel sentence pair $(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x} = (x_1, ..., x_m)$ is the source sentence, $\boldsymbol{y} = (y_1, ..., y_n)$ is the reference target sentence, we would like to construct a target word w and its corresponding target context $\boldsymbol{c} = (\boldsymbol{c}_l, \boldsymbol{c}_r)$ and human-typed characters $\boldsymbol{t}$, where the translation pieces $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$ are on the left and right side of the target word w.

According to Li et al. (2021), we first randomly sample a target word $w = \boldsymbol{y}_t$, and then we sample four types of context types:

- Zero-context: both $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$ are empty;
- Prefix: randomly sample a translation piece $\boldsymbol{c}_l = \boldsymbol{y}_{i:j}$ from $\boldsymbol{y}$, where $i < j < t$. The $\boldsymbol{c}_r$ is empty.
- Suffix: randomly sample a translation piece $\boldsymbol{c}_r = \boldsymbol{y}_{i:j}$ from $\boldsymbol{y}$, where $t < i < j$. The $\boldsymbol{c}_l$ is empty.
- Bi-context: sample $\boldsymbol{c}_l$ as in prefix, and sample $\boldsymbol{c}_r$ as in suffix.

Last but not least, we need to generate human-typed characters $\boldsymbol{t}$ for the target word w, we adopt a heuristic method - we randomly sample a position i in the target word w, where $0 < i < |w|$, and simulate human-typed characters $\boldsymbol{t} = w_{1:i}$. For languages like Chinese, the human input is the phonetic symbols of the word, we use pypinyin[3] to implement this conversion. So far, we get the tuple $(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}, w)$, which can be viewed as a simulated training example for the WLAC shared task.

2.3 Vocabulary

Considering that WLAC is a word-level task, we don't use tools to do any subword segmentation. We directly use Moses scripts[4] to tokenize English and German sentences, and jieba[5] for Chinese sentences. The vocabulary size for each language direction is shown in Table 2.

3 Word-Level AutoCompletion Systems

In this section, we mainly introduce the *Generate-then-Rerank* framework. Both the generator and the reranker's architecture are based on Transformer(Vaswani et al., 2017) with the modification that the decoder is bi-directional to leverage more context information. It is important to note that we borrow the idea from Li et al. (2021) that we view WLAC as a word prediction task and *only use human-typed characters* $\boldsymbol{t}$ *as hard constraints*.

3.1 Model Architecture: Transformer

The vanilla Transformer (Vaswani et al., 2017) adopts a sequence-to-sequence architecture consisting of an encoder and a decoder. Specifically, the encoder is a stack of L encoder blocks and each block consists of a multi-head self-attention module and a feed-forward network (FFN). The decoder is also a stack of L decoder blocks, the main differences between the Transformer encoder and Transformer decoder are mainly reflected in two aspects: First, in each decoder block, there is an additional cross-attention module between the multi-head self-attention module and the feed-forward network. Second, the multi-head self-attention modules in the decoder are uni-directional while they are bi-directional in the encoder.

In the neural machine translation task setting, given a source sentence $\boldsymbol{x}$ and a target sentence $\boldsymbol{y}$, the decoder generates $\boldsymbol{y}$ as:

$$P(\boldsymbol{y}|\boldsymbol{x};\theta) = \prod_{t=1}^{|\boldsymbol{y}|} P(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x};\theta) \qquad (1)$$

[3] https://github.com/mozillazg/python-pinyin

[4] https://github.com/moses-smt/mosesdecoder

[5] https://github.com/fxsjy/jieba

Thus, the Transformer model is typically trained by minimizing the cross entropy:

$$\mathcal{L}_{NMT} = -\sum_{t=1}^{|\boldsymbol{y}|} \log P(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x}; \theta) \quad (2)$$

Since Transformer is designed for auto-regressive generation tasks, we cannot directly adopt it to the WLAC task, which is essentially a natural language understanding task. Inspired by the successful practice of Conditional Masked Language Modeling (Ghazvininejad et al., 2019) in non-autoregressive machine translation, we take the same idea to train our model for the WLAC shared task.

Bi-directional Decoder Our decoder's architecture is roughly the same as the standard Transformer decoder except that the multi-head self-attention sub-layer. The standard Transformer decoder can only attend the left-side target context, while in our model, it can attend to all target words and make use of both left-side and right-side context information to better predict the <*mask*> token.

3.2 Generator

Span-Level Mask Prediction The primitive object function for a simulated training example in Generator is as follows:

$$\mathcal{L}_G = -\log P(w|\boldsymbol{x}, \boldsymbol{c}; \theta_G) \quad (3)$$

In our preliminary experiments, we find that it is hard to train the generator because, in every mini-batch, a simulated training example provides *only one* training signal, which makes the model easy to overfit. The importance of the density of training signals has been discussed in the Pretrained Language Model(Clark et al., 2020). To this end, we adopt an efficient sampling approach —— Span-Level Mask Prediction. As described in section 2.2, once we get the tuple $(\boldsymbol{x}, \boldsymbol{c})$, we use it to predict all the missing words in the masked span between $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$. In the Pretrained Language Model, Joshi et al. (2020) has adopted the same idea as in our work. But one major difference is that, unlike Joshi et al. (2020), we have to set the position id of the masked word to be the same; otherwise, there will be a large gap between the training stage and the inference stage.

3.3 Reranker

So far, we have modeled the WLAC task as a classification task, that is, an extreme classification task. Inspired by recent works to introduce label knowledge to enhance text representation (Yang et al., 2021; Ma et al., 2022), we propose to use a generator-reranker framework to solve the WLAC task. We use the generator to recall positive and negative labels and use a reranker to distinguish positive labels from these labels. Specifically, we use the same Transformer architecture as the generator. But the reranker's input and objective function are different from the generator.

Input We obtain top-K labels $\mathcal{W} = \{w_1, w_2, ..., w_K\}$ through ranking the scores generated by the generator. Then, for each candidate label w_i in $\mathcal{W}$, we replace the <*mask*> token with w_i. So the input tuple becomes $(\boldsymbol{x}, \boldsymbol{c}, w_i)$. And the multi-class classification head of the original decoder becomes a binary classification head, which is used to measure whether the candidate label w_i matches the source sentence and target context.

Objective Function The objective function is as follows.

$$\mathcal{L}_R = \begin{cases} -\log P(w_i, \boldsymbol{x}, \boldsymbol{c}; \theta_R), & \text{if } w_i = w \\ -(1 - \log P(w_i, \boldsymbol{x}, \boldsymbol{c}; \theta_R)), & \text{otherwise.} \end{cases} \quad (4)$$

3.4 Model Configuration

We implement our models with Fairseq toolkit(Ott et al., 2019)[6]. Our models follow the Transformer-Base architecture(Vaswani et al., 2017), the key model architecture configurations and training configurations are listed in Table 4 and Table 5. Each model is trained on 8 NVIDIA Tesla V100 GPUs, each of which has 32GB memory.

4 Experimental Results

We report experimental results in four language directions: Chinese ⇒ English, English ⇒ Chinese, German ⇒ English and English ⇒ German. Table 3 shows the main experimental results on the official test sets with automatic accuracy evaluation and human accuracy evaluation.

[6] https://github.com/facebookresearch/fairseq

#	Systems	Zh⇒En		En⇒Zh		De⇒En		En⇒De	
		Auto	Human	Auto	Human	Auto	Human	Auto	Human
1	Generator	54.05	85.00	53.98	83.25	57.27	78.75	41.82	55.50
2	Reranker	51.11	83.75	48.90	77.50	54.32	76.25	40.69	53.50

Table 3: The main results of different systems in four language directions. The results are the averaged automatic accuracy and human accuracy on four types of translation context (i.e., zero context, prefix, suffix, and bi-context).

Configuration Name	Configuration Value
encoder layers	6
decoder layers	6
attention heads	8
word embedding dim	512
FFN embedding dim	2048
hidden dim	512
dropout	0.1
attention dropout	0.0
activation droupout	0.0
Pre-LN	False
share decoder input output embed	True

Table 4: The exact specifications of the Transformer we adopt.

Configuration Name	Configuration Value
number of training steps	10000
update freq	1
learning rate scheduler	inverse sqrt
warmup updates	4000
warmup init learning rate	1e-7
learning rate (generator)	5e-3
learning rate (reranker)	1e-3
max tokens per batch	32k
optimizer	Adam

Table 5: Training configuration for our generator model and reranker model.

The performance of the generator is as expected, and as demonstrated in Li et al. (2021), without using the bi-directional decoder, the generator performs relatively poorly. Additionally, we conduct an ablation study on Chinese ⇒ English subtask to demonstrate the effectiveness of the span-level mask prediction, the model without leveraging the span-level mask prediction strategy performs poorly, with a drop of -10.1 in accuracy on the validation set.

However, the performance of the reranker is not as expected. We conjecture that this is due to the insufficiency of the training procedure of the reranker. Initializing reranker's weights with generator's weights or with PLM's weight will boost the performance of reranker, we leave this as future work.

5 Conclusion

This paper describes the IIGROUP's systems submitted to the Word-Level AutoCompletion task at WMT22. We adopt a *Generate-then-Rerank* framework. The experimental results demonstrate the effectiveness of the generator.

However, due to the lack of computing power and time, the results of our experiments don't show the effectiveness of our reranker. We discuss this issue in section 4 and we will try to solve this in future work.

References

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, et al. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121.

Guoping Huang, Lemao Liu, Xing Wang, Longyue Wang, Huayang Li, Zhaopeng Tu, Chengyan Huang, and Shuming Shi. 2021. Transmart: A practical interactive machine translation system. *arXiv preprint arXiv:2105.13072*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Huayang Li, Lemao Liu, Guoping Huang, and Shuming Shi. 2021. Gwlan: General word-level autocompletion for computer-aided translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4792–4802.

Jie Ma, Miguel Ballesteros, Srikanth Doss, Rishita Anubhai, Sunil Mallya, Yaser Al-Onaizan, and Dan Roth. 2022. Label semantics for few shot named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1956–1971.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Sebastin Santy, Sandipan Dandapat, Monojit Choudhury, and Kalika Bali. 2019. Inmt: Interactive neural machine translation prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 103–108.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yanling Xiao, Lemao Liu, Guoping Huang, Qu Cui, Shujian Huang, Shuming Shi, and Jiajun Chen. 2022. Bitiimt: A bilingual text-infilling method for interactive machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1958–1969.

Pan Yang, Xin Cong, Zhenyu Sun, and Xingwu Liu. 2021. Enhanced language representation with label knowledge for span extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4623–4635.

HW-TSC's Submissions to the WMT22 Word-Level Auto Completion Task

**Hao Yang, Hengchao Shang, Zongyao Li, Daimeng Wei, Xianghui He,
Xiaoyu Chen, Zhengzhe Yu, Jiaxin Guo, Jinlong Yang
Shaojun Li, Yuanchang Luo, Yuhao Xie, Lizhi Lei, Ying Qin**
Huawei Translation Service Center, Beijing, China
{yanghao30, shanghengchao,lizongyao,weidaimeng,hexianghui,
chenxiaoyu35,yuzhengzhe,guojiaxin1,yangjinlong7,
lishaojun18,luoyuanchang,xieyuhao2,leilizhi,qinying}@huawei.com

Abstract

This paper presents the submissions of Huawei Translation Services Center (HW-TSC) to WMT 2022 Word-Level AutoCompletion Task. We propose an end-to-end autoregressive model with bi-context based on Transformer to solve the current task. The model uses a mixture of subword and character encoding units to realize the joint encoding of human input, the context of the target side and the decoded sequence, which ensures full utilization of information. We use one model to solve four types of data structures in the task. During training, we try using a machine translation model as the pre-trained model and fine-tune it for the task. We also add BERT-style MLM data at the fine-tuning stage to improve model performance. We participate in zh→en, en→de, and de→en directions and win the first place in all the three tracks. Particularly, we outperform the second place by more than 5% in terms of accuracy on the zh→en and en→de tracks. The result is buttressed by human evaluations as well, demonstrating the effectiveness of our model.

1 Introduction

In recent years, machine translation quality has improved significantly with advances in model architecture (Sutskever et al., 2014; Bahdanau et al., 2014; Gehring et al., 2017; Vaswani et al., 2017), bilingual data availability, as well as data augmentation strategies (Liu et al., 2016; Freitag et al., 2017; Johnson et al., 2017; Zhang et al., 2018; Edunov et al., 2018; Wu et al., 2019; Li et al., 2019). In scenarios where machine translation is used to facilitate understanding, machine translation outputs can basically satisfy audience's demands. However, in areas where translation quality is crucial (such as translating product manual, patent description, etc.), post-editing is required. Techniques to improve post-editing efficiency are meaningful and necessary. Researches (Barrachina et al., 2009; Green et al., 2014; Knowles and Koehn, 2016; Santy et al., 2019) in this regard falls into this category of computer-aided translation (CAT).

Word-Level auto Completion, as a new task in WMT22, fall into this category as well. This task aims at auto-completing a target word given a source sentence, translation context, and a human-typed character sequence, so as to improve post-editing efficiency. Li et al. (2021) define the task in detail, offer comprehensive analysis and provide a baseline system.

For this task, we choose the subword-level modeling strategy (Kudo and Richardson, 2018). Comparing with word-level modeling, this strategy enables the usage of pre-trained models from other mainstream tasks, and solves the out-of-vocabulary (OOV) issues at the same time. As human-typed input is just several characters of the target word, the input is not suitable for subword segmentation. We use character-level encoding instead. Our final submission is an autoregressive model with bi-context, ensuring mixed encoding of characters and subwords.

In view of the possible discontinuity between the context and human-typed inputs in the target-side text, we use tags to wrap the inputs, and then encode jointly with the context, in conjunction with the autoregressively decoded pre-token sequence. The joint coding maximizes the usage of information without introducing additional RNN (Cho et al., 2014) or vocabulary reduction modules.

During training, we use a standard machine translation model as the pre-trained model, and fine-tune it for this task. We then add BERT-style Mask Language Model (MLM) (Devlin et al., 2018) data in the fine-tuning stage to enhance the language model capabilities of the decoder, thereby improving the overall model performance.

The inference mechanism is different from that of traditional NMT. In general, the entire decoder sequence must be used for encoding each token,

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1192 - 1197
December 7-8, 2022 ©2022 Association for Computational Linguistics

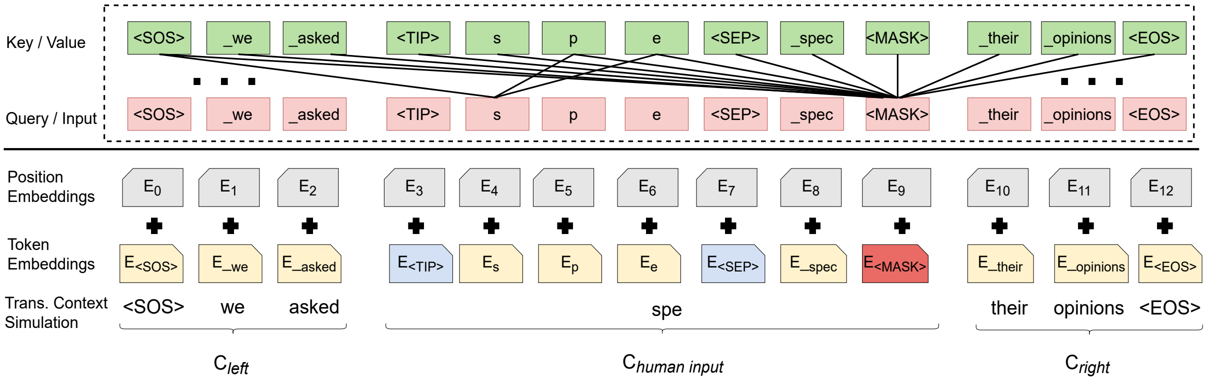


Figure 1: The input representation of our model's decoder. C_{left} and C_{right} are the context. E_s, E_p, E_e are the char embedding of human input "spe". <TIP> is the separator for human input and left context. <SEP> is the separator for human input and decoded sequence. <MASK> represents the potenial target word in this translation context.

which reduces the inference performance to a certain extent. However, given the model's parallel ability and the short decoding sequence (the number of subwords in a word), this issue is not very serious for this task and this strategy is applicable for practical use. Figure 1 shows the input representation of our model's decoder.

We submit results for three language directions. All of them achieve the highest accuracy. Our Zh→En and En→De submissions even outperform the second place by 5% in terms of accuracy, and get a good lead in human evaluation, demonstrating the effectiveness of our strategy.

2 Data Processing

Zh→En data for this task comes from UN V1.0 (about 15.9M) and En↔De data comes from WMT14 (about 4.5M). The task allows the use of additional monolingual data, but we add no additional data on the zh-en track given the amount of bilingual data available. An additional 24M monolingual data is used for the En-de track, and the data comes from the WMT news task as well. we generate synthetic parallel data by sampling BT (Edunov et al., 2018) for the En→De track and by beam BT for the Dn→En track. The specific reasons will be given later.

We follow basic strategies to cleanse the data, including: deduplication, garbled character filtering, XML conversion, and fast-align(Dyer et al., 2013), etc. The data sizes before and after data processing are shown in table 1.

As for subword, we employ sentencpiece on Zh→En track, and set the vocabulary size to 36k. On En→De track, we use the BPE algorithm, and set the vocabulary size to 32K.

Lang-Pair	Origin	Cleaned
Zh-En	15.9M	15.5M
En-De	4.5M	4.3M

Table 1: Overview of training para data.

left-context ***<tip> t i p s <tip>*** *right-context*

Figure 2: The joint encoding of context and human input.

3 System design

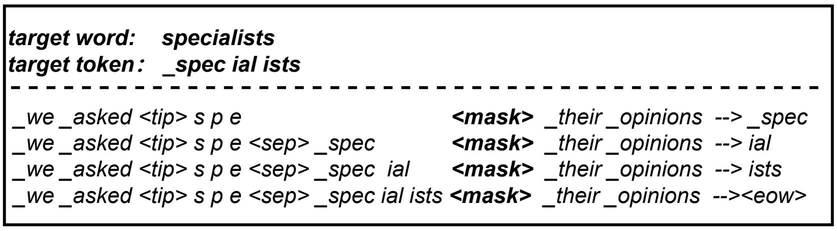
In this chapter, we introduce the model structure, training strategy, inference strategy and corresponding data generation strategy used for this task. Our model is based on the encoder-decoder architecture of the standard Transformer.

3.1 Model Structure

We use Transformer as our model architecture. For convenience, we only use a 25 encoder layers and 6 decoder layers deep model. The parameters of the model are the same as Transformer-big. We just change the post-layer-normalization to the pre-layer-normalization (Sun et al., 2019), and increase the number of encoder layers to 25.

3.2 Modeling Units

In general, we use a mixed encoding strategy that encoding subword-level and character-level information at the same time. To be more specific, the model conducts subword-level encoding on source and target context information, and character-level encoding on human-typed input, as it is just several characters of a word. Apparently, the model can



```
target word:    specialists
target token:   _spec ial ists
-----------------------------------------------------------------------
_we _asked <tip> s p e                           <mask>  _their _opinions  --> _spec
_we _asked <tip> s p e <sep> _spec               <mask>  _their _opinions  --> ial
_we _asked <tip> s p e <sep> _spec  ial          <mask>  _their _opinions  --> ists
_we _asked <tip> s p e <sep> _spec ial ists <mask>  _their _opinions  --><eow>
```

Figure 3: The process of data generation.

encode the two types of information at the same time.

3.3 Joint Encoding

In the above chapter, we discussed our modeling granularity for context and human-typed input. According to the settings of the task, context information and human-typed input information may be discontinuous. Here we insert the tag <tip> before and after the tip to wrap the human-typed input to distinguish the two. Schematic diagram is shown in Figure 2.

The context and human-typed input of the translation test can be jointly encoded, which ensures the maximum usage of information.

3.4 Autoregressive Decoding with Bi-context

In the task, there are four types of data: left-context, right-context, zero-context and bi-context. If we use four models to process these four types of data, the problem can be solved, but the task will be complicated. Instead, we can regard the first three types as special cases of the last type, so we directly design an autoregressive decoding strategy with bi-context, and use a single model to process all types of data.

To be more specific, decoding is performed with Mask token as the anchor point. The encoder encodes the source-side text. The mask, in conjunction with context and human-typed input, is encoded at the decoder side to predict the first subword of the target word. The first subword (pre-token) will be encoded as well. Then the model continues to use the mask for decoding until a complete word is decoded. The overall architecture diagram of the model decoder is shown in figure 1.

Here are two points: 1. The mask token replaces the second tip described in the previous section. The mask token is capable of distinguishing the human-typed input from other information and masking at the same time. 2. The newly added <sep> tag is responsible for distinguishing between human-typed input and decoded pre-token information.

3.5 Data Generation

Based on our previous coding strategy for various types of information, we first use the script provided by the organizer to generate word-level training data. Then, we use the subword-level model to perform subword processing on the source text and translation context. Regarding the target word, we add a tag <eow> at the end of word after subword segmentation. Assuming that the number of subwords is N, we generate N sets of training data to simulate the entire autoregressive process. We call the data as WLAC (Word Level Auto Completion) data to distinguish it from terms. A case study of the generation is show in Figure 3.

In the process of generating data, we also add some rules to improve efficiency. We remove sentences with too short target words or too long human-typed input by a given probability. In addition, we keep only 1/10 of the training samples of eos that predict the end of the sentence.

In order to effectively validate the performance of the model during training, we generate a test set using the same strategy. WMT19 news test is used for the Zh→En track, and WMT14 new-test is used for En↔De tracks. We do not use a filtering strategy when generating the test set.

4 Experiment

During the experiment, we first build a baseline based on the MT model in order to measure our model's performance more accurately. After that, in the training process, we adopt several strategies to improve the model performance, that is, fine-tuning a MT model and introducing BERT-style MLM data. Validation and debugging of these strategies are done on the Zh→En track. We use the finally determined strategy to train models for other tracks.

Lang-Pair	Baseline	MT-tune	Mix-tune	Average	Ensemble	WMT22
zh-en	62%	74%	77.19%	78.69%	78.96%	59.40%
en-de	-	dvivied	81.79%	82.86%	82.80%	62.06%
de-en	-	-	77.83%	79.05%	79.77%	63.82%

Table 2: The main results of our experiments. MT-tune refers to using WLAC data to fine-tune a standard MT model. MLM-Mix-tune refers to using WLAC and BERT-style MLM data to fine-tune the MT model.

4.1 Baseline based on the MT Model

First, we consider whether the current task can be solved by directly relying on the outputs by an MT model trained with bilingual data. By doing so, we lower the requirements of this strategy and regard a case as correct as long as the predicted word appears in the MT result.

We obtain an accuracy of 62.5% on the Zh→En track by using the above-mentioned approach. We use this as a benchmark for optimization. If our designed strategy cannot exceed this level of accuracy, the strategy fails.

4.2 MT Model Finetune

After obtaining the baseline MT model, we then fine-tune it using the generated WLAC data. It should be noted that the self-attention layer of the standard NMT model's decoder does not have the ability to generate attention to the right, and our decoder is a mask-based prediction model, so we need to break this limitation. This is also a gap between the two tasks.

4.3 BERT-Style MLM Data Fine-tune

In the fine-tuning stage on the basis of a MT model, through analyzing each type of data, we find that the the accuracy of prefix is higher than that of the suffix. We assume this is because there is no right-side information during the training a pre-trained NMT model. As a result, the model may learn right-context less efficiently than the left-context.

According to the task setting, the context of the target side is an incomplete fragment and is given randomly. At the same time, tips are not necessarily continuous, so the overall translation is relatively confusing. Source-side information is important so we deepen the number of encoding layers. In addition, using the mask as the decoding anchor causes the decoder to change from the standard language model mode to the mask language model mode. To address these issues, we add a same proportion of BERT-style MLM data with source-side information. Given the availability of original text, we enlarged the probability of the mask to 25%.

4.4 Average and Ensemble

Finally, we adopt commonly used strategies to improve the model performance, including averaging and ensemble, and we find that both of the strategies lead to performance improvement. Particularly, averaging brings significant improvement.

5 Result and Analysis

Due to time restriction, we only conduct detailed comparison experiments on the Zh→En track. En↔De tracks simply follows the final strategy we apply to the Zh→En track. The results are shown in Table 2.

First of all, the performance of the MT baseline we trained is very poor, indicating that the MT task is not well adapted to the current task. So we give up the idea of using the MT results to enhance the model performance.

After our constructed WLAC data is added, the model performance improves by nearly 12 points, indicating the effectiveness of our strategy. But it is worth noting that the En→De model does not converge after adding the ST/BT data. We assume that the quality of the ST data is not good. In addition, the target-side of WLAC data is confusing, resulting in training failure. So for the De→En model, we directly generate BT data based on beam search to avoid the issue.

After MLM data is added, we again obverse a significant improvement. The accuracy on the Zh→En reaches 77.19%. The En→De model, which was not converged at the previous stage, gains an accuracy of 81.79%. The results support our assumption that MLM data can enhance the language model ability of the decoder, while avoiding noise interference from the source-side text.

In the end, model averaging leads to improvements on all three tracks, and the improvement is more significant than ensemble. Ensemble leads to significant improvement on the De→En track but limited improvement on the other two tracks. We assume this is because the De→En model is not sufficiently trained due to time restriction.

6 Conclusion

In this paper, we detail our team's participation in the WMT22 word-level AutoComplete task. We first analyze the input and output, as well as challenges in this task. We notice that the modeling granularity of human-typed input and context information are different. Therefore, we propose modeling human-typed input at character-level and modeling context information at subword-level, explicitly distinguishing and jointly encoding the two, thereby maximizing information usage in the encoding stage. At the same time, there is a semantic discontinuity between context and human-typed input. We add tags to differentiate the two. Finally we propose an autoregressive model with bi-context to process four types of data at the same time. During training, we use an NMT model as the pre-trained model and fine-tune it for this task. BERT-style MLM data is also introduced to improve the model performance, and at the same time to solve the single-direction decoding issue of the self-attention model. In the end, our models are well adapted to the task and gain safe leads in both automatic and human evaluations.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, et al. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 644–648.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 489–500.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *CoRR*, abs/1702.01802.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR.

Spence Green, Sida I Wang, Jason Chuang, Jeffrey Heer, Sebastian Schuster, and Christopher D Manning. 2014. Human effort and machine learnability in computer aided translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1225–1236.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Rebecca Knowles and Philipp Koehn. 2016. Neural interactive translation prediction. In *AMTA (1)*, pages 107–120.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The niutrans machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 257–266.

Huayang Li, Lemao Liu, Guoping Huang, and Shuming Shi. 2021. Gwlan: General word-level autocompletion for computer-aided translation. *arXiv preprint arXiv:2105.14913*.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *Proceedings of the 2016 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 411–416.

Sebastin Santy, Sandipan Dandapat, Monojit Choudhury, and Kalika Bali. 2019. Inmt: Interactive neural machine translation prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 103–108.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 374–381.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4205–4215.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

TSMind: Alibaba and Soochow University's Submission to the WMT22 Translation Suggestion Task

Xin Ge[1]*, **Ke Wang**[1]*, **Jiayi Wang**[1], **Nini Xiao**[1,2], **Xiangyu Duan**[2], **Yu Zhao**[1], **Yuqi Zhang**[1]†
[1]Alibaba Group Inc. [2]Soochow University
{shiyi.gx,moyu.wk,joanne.wjy}@alibaba-inc.com
{nnxiaonnxiao,xiangyuduan}@suda.edu.cn, {kongyu, chenwei.zyq}@alibaba-inc.com

Abstract

This paper describes the joint submission of Alibaba and Soochow University, TSMind, to the WMT 2022 Shared Task on Translation Suggestion (TS). We participate in the English ↔ German and English ↔ Chinese tasks. Basically, we utilize the model paradigm fine-tuning on the downstream tasks based on large-scale pre-trained models, which has recently achieved great success. We choose FAIR's WMT19 English ↔ German news translation system and MBART50 for English ↔ Chinese as our pre-trained models. Considering the task's condition of limited use of training data, we follow the data augmentation strategies proposed by Yang et al. (2021) to boost our TS model performance. The difference is that we further involve the dual conditional cross-entropy model and GPT-2 language model to filter augmented data. The leader board finally shows that our submissions are ranked first in three of four language directions in the Naive TS task of the WMT22 Translation Suggestion task.

1 Introduction

Computer-aided translation (CAT) (Barrachina et al., 2009; Green et al., 2014, 2015; Knowles and Koehn, 2016) has become more and more popular to help increase the quality of machine translation (Lopez, 2008; Koehn, 2009) result. It also improves the efficiency of translators by combining the results of machine translation and the content edited by translators in the process of translation or post-editing (Bowker, 2002; Lengyel and Ugray, 2004; Bowker and Fisher, 2010; Bowker, 2014; Chatterjee, 2019).

Post-editing based on machine translation is typical in CAT. Recent works (Domingo et al., 2016; González-Rubio et al., 2016; Peris et al., 2017) propose interactive protocols and algorithms so that humans and machines can collaborate during translation, and machines can automatically provide feedback on humans' edits. One interesting mode is Translation Suggestion (TS) (Yang et al., 2021), which offers alternatives for specific spans of words in the generated machine translation. It will be convenient if the model refines translation results in those specified locations with potential translation errors. Yang et al. (2021) released a benchmark dataset for TS, *WeTS*, which is one of the shared tasks in WMT22. At the same time, they proposed an end-to-end Transformer-like model for TS as the benchmark system.

However, the lack of many labeled TS data limits the training of a large Transformer model to some extent. Though Yang et al. (2021) have tried to utilize XLM-Roberta (Conneau et al., 2019) to initialize the encoder of the Transformer, the decoder has to be trained from scratch, which leads to relatively low BLEU scores for some specific TS spans. We investigate the potential of other encoder-decoder pre-trained models by experiments to see if there is still room for improvement. Finally, we have found that pre-trained Transformer NMT models could be suitable choices to be fine-tuned with the limited size of TS data. In addition, we applied similar data augmentation strategies proposed in Yang et al. (2021), but use the well-trained alignment models between source and target languages from Lu et al. (2020) to filter out high-quality augmented data. Our submissions are ranked first in three of four language directions in the WMT22 Translation Suggestion task.

2 The Model

We train a simple end-to-end Transformer model for each language pair to generate the translation suggestion candidates. The source sentence and the masked translation, in which an incorrect span requiring an alternative has been replaced with a special mask tokens in advance, are concatenated with a special separation token *[SEP]*. Afterward,

[*] indicates equal contribution.
[†] indicates the corresponding author.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1198 - 1204
December 7-8, 2022 ©2022 Association for Computational Linguistics

Symbol	Definition
$\mathbf{x}$	Sentence in source language
$\mathbf{y}$	Machine translation result of $\mathbf{x}$
$\mathbf{r}$	Reference sentence $\mathbf{x}$
$\mathbf{x}^i$	The i-th token of x
$\|\mathbf{x}\|$	Length of $\mathbf{x}$, i.e. the number of tokens in $\mathbf{x}$
$\mathbf{x}^{i:j}$	The fragment of $\mathbf{x}$ from position i to j
$\mathbf{x}^{\neg i:j}$	The masked version of $\mathbf{x}$, in which tokens at the position from i to j of x is replaced with a mask token.
$\hat{\mathbf{p}}$	All aligned-phrase pair between $\mathbf{y}$ and $\mathbf{r}$, pair look likes $(\mathbf{y}^{i:j}, r^{a:b})$
$\hat{\mathbf{y}}$	Replace $\mathbf{y}^{i:j}$ with $\mathbf{r}^{a:b}$ in $\mathbf{y}$, and get another new sentence $\hat{y}$

Table 1: Notations

	WMT22	Filter Length	Filter Quality
en-zh	23.2M	9.78M	6.9M
en-de	30.0M	12.73M	8.18M

Table 2: Number of parallel samples remained after filtering by length and cross-entropy quality score (Lu et al., 2020).

we feed the concatenated sequence as input of the Transformer encoder and the translation suggestion needs to be generated by the Transformer decoder. The model is trained in the same way of a normal translation model.

Considering that the TS task also relies on alignments of hidden representations between the source and the target language, a well-trained translation model can be a good starting point for TS model training. The weights of our model are initialized with a pre-trained Transformer NMT model. Then, a two-phase training pipeline is applied. In the first phase, the model is trained with pseudo corpus derived from data augmentation described in Section 3. In the second phase, we fine-tune the model with the real TS train data released by the organizers.

3 Data Augmentation

We follow the data augmentation methods provided by (Yang et al., 2021) to generate three types of pseudo data for TS model training: data sampled on the golden parallel corpus, data sampled on the pseudo parallel corpus, and data extracted with word alignment. However, the details of the pseudo data augmentation in this paper are slightly different from those of Yang et al. (2021). Full details are exhibited in the following subsections.

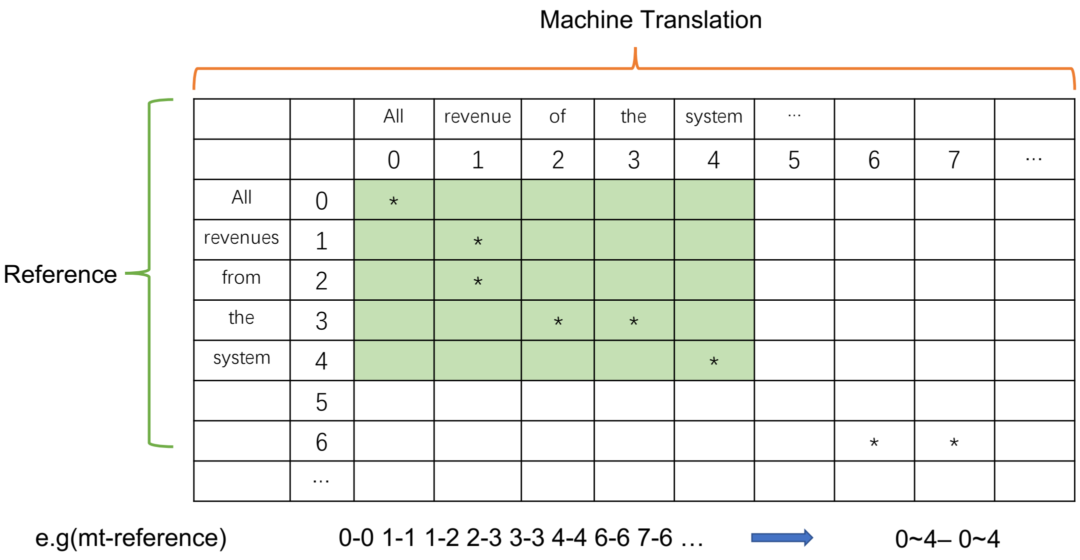
Algorithm 1 Algorithm of Phrase Align

```
Input: y, r, A
Output: p̂
Function GenerateAlign(y, r, A):
    $yt = size(\mathbf{y}), rt = size(\mathbf{r})$
    for $i \leftarrow 0$ to $yt$ do
        for $j \leftarrow i$ to $yt$ do
            for $a \leftarrow 0$ to $rt$ do
                for $b \leftarrow a$ to $rt$ do
                    if IsMatch(y, r,i, j, a, b, A) then
                        do
                            $i += 1; a += 1$
                        while $\mathbf{y}^i == \mathbf{r}^a$
                        do
                            $j -= 1; b -= 1$
                        while $\mathbf{y}^j == \mathbf{r}^b$
                        $\hat{\mathbf{p}}.add((\mathbf{y}^{i:j}, \mathbf{r}^{a:b}))$
    return $\hat{\mathbf{p}}$
Function IsMatch(y, r, i,j,a,b,A):
    for $ii \leftarrow i$ to $j$ do
        let T = $\{t_i | \mathbf{r}^{t_i}$ is aligned with $y^{ii}$ in $\mathbf{A}$ $\}$
        foreach $t_i \in T$ do
            if $t_i < a$ or $t_i > b$ then
                return False
    for $aa \leftarrow a$ to $b$ do
        let T = $\{t_a | \mathbf{r}^{aa}$ is aligned with $y^{t_a}$ in $\mathbf{A}$ $\}$
        foreach $t_a \in T$ do
            if $t_a < i$ or $t_a > j$ then
                return False
    return True
```

3.1 Sampling from golden parallel corpus

Raw parallel corpus is firstly filtered by the sentence length. All sentence pairs that have less than 20 words or more than 80 words on any side are removed.

Considering that there might be noise data in the corpus, we apply the dual conditional cross-entropy model (Lu et al., 2020) to obtain a quality score for each sample. Sentence pairs with low quality are



		All	revenue	of	the	system	…			
		0	1	2	3	4	5	6	7	…
All	0	*								
revenues	1		*							
from	2		*							
the	3			*	*					
system	4					*				
	5									
	6							*	*	
	…									

Figure 1: In this example, we have the alignment info between machine translation (MT) and reference sentences: 0-0, 1-1, 1-2, 2-3, 3-3 4-4, 6-6, 6-7, the phrase from $0 \sim 4$ in MT are aligned to $0 \sim 4$ in reference. The rectangle enclosed by the aligned phrases between MT and reference should satisfy that each row and each column has at least one *.

filtered.

Then we generate a pseudo corpus with the remained high-quality parallel corpus. ($\mathbf{x}$, $\mathbf{r}$) is marked as the sentence pair of the parallel corpus, where $\mathbf{x}$ is the source sentence and $\mathbf{r}$ is the golden reference. $\|\mathbf{r}\|$ represents the number of tokens in $\mathbf{r}$.

The first step is to randomly sample the length l to mask for the reference r from a uniform distribution:

$$l \sim U(1, \|\mathbf{r}\|) \tag{1}$$

Then a span with l tokens $\mathbf{r}^{i:j}$ is randomly selected by:

$$i \sim U(0, \|\mathbf{r}\| - l), \;\; j = i + l \tag{2}$$

Finally, we get the TS training data $(\mathbf{x}, \mathbf{r}^{\neg i:j}, r^{i:j})$ from each parallel sentence pair $(\mathbf{x}, \mathbf{r})$, where $\mathbf{r}^{\neg i:j}$ is denoted as the masked version of r, in which $\mathbf{r}^{i:j}$ is replaced with a mask token, e.g <MASK_REP>.

3.2 Sampling on Pseudo Parallel Corpus

In addition, the monolingual corpus is another source for data augmentation. We first filter the monolingual data with a language identification process. Then pseudo parallel corpus is generated with NMT models. Finally, TS training data can be generated as we do in Section 3.1.

3.3 Extracting with Word Alignment

In the task of TS, the labels for the masked span is always correct while the translation contexts of the span, $\mathbf{y}^{\neg i:j}$ are not error-free. Therefore, both of the above two types of pseudo data are biased from the task. In pseudo data sampled from golden parallel corpus, the translation contexts are error-free. And the labels of pseudo data from machine translation results are not always correct. To reduce the bias, another way of data augmentation is proposed in Yang et al. (2021). They utilize the alignment between the machine translation and the golden reference to generate pseudo-training samples for TS. We use the similar idea and the details of our alignment-based data augmentation algorithm are described as follows.

Given the triplet ($\mathbf{x}$, $\mathbf{y}$, $\mathbf{r}$) where $\mathbf{x}$ is the source sentence, $\mathbf{y}$ is the machine translation result generated by NMT models, and $\mathbf{r}$ is the reference, we need to find aligned segment pairs $(\mathbf{y}^{i:j}, \mathbf{r}^{a:b})$ between $\mathbf{y}$ and $\mathbf{r}$.

First, we use the Fast Align toolkit (Dyer et al., 2013) to extract token alignments between $\mathbf{y}$ and $\mathbf{r}$. The align result $\mathbf{A}$ is a list of aligned indexes in the format of i-a, which means token $\mathbf{y}^i$ is aligned to $\mathbf{r}^a$. With the token alignments, the next step is to extract aligned-phrase pairs, denoted as $\hat{\mathbf{p}}$. Figure 1 shows an example of an aligned phrase between MT and reference. The algorithm of the aligned-

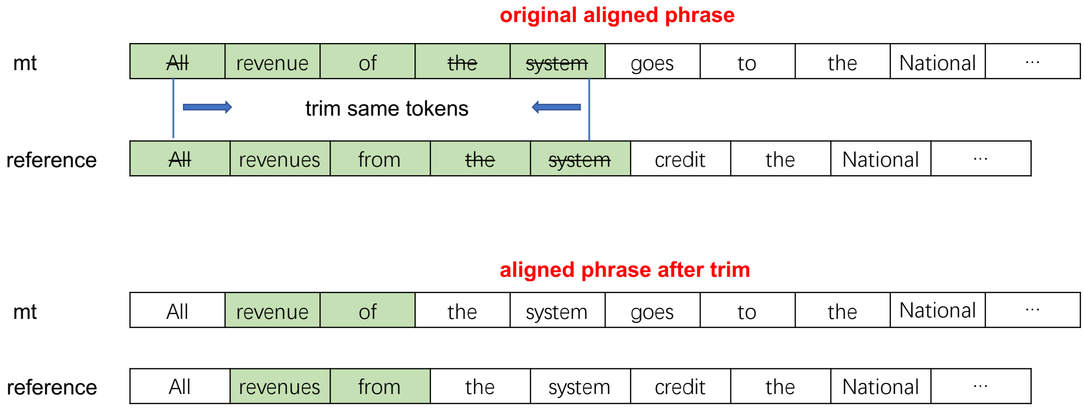


Figure 2: As shown in Figure 1, we get the original aligned phrase between MT and reference which are "All revenue of the system" and "All revenues from the system". We then trim the tokens that appear in both MT and reference to compress the aligned phrase. Finally, we get the trimmed aligned phrase: "revenue of" and "revenues from"

Method	En-De	De-En	En-Zh	Zh-En
TSMind	45.90	43.37	30.21	28.77
-w/o first-phase training	37.14	33.23	21.20	16.44
-w/o second-phase training	37.37	36.83	21.84	19.19

Table 3: Sacre-BLEU on the validation sets of Sub-Task 1 (Naive TS) of the WMT'22 Translation Suggestion Task.

phrase extraction is presented in Algorithm 1 from line 1 to line 13. The aligned phrases are a subset of SMT's phrase extraction (Koehn et al., 2003) with two restricts. 1) Each row and each column of a aligned phrase has at least one token aligned (a * in Figure 1); 2) We take only the longest phrase and the sub-phrases are not taken. After the original aligned phrase is obtained, we remove tokens that appear in both MT and reference to get the trimmed result as shown in Figure 2. We trim these common tokens because we want the model to focus more on the incorrect spans and its alternatives. The pseudo-code of the phrase-alignment is presented in the Algorithm 1. We denote the aligned phrase as $\mathbf{y}^{i:j}$ and $\mathbf{r}^{a:b}$, $\mathbf{y}^{\neg i:j}$ represents the masked version of $\mathbf{y}$ as described in Section 3.2.

Now we need to judge whether $\mathbf{r}^{a:b}$ is better than $\mathbf{y}^{i:j}$ in the context of $\mathbf{y}^{\neg i:j}$. We replace $\mathbf{y}^{i:j}$ with $\mathbf{r}^{a:b}$ in $\mathbf{y}$, and get another new sentence $\hat{\mathbf{y}}$. First, we use the dual conditional cross-entropy model as described in Section 3.1 to calculate the quality score of $(\mathbf{x}, \hat{\mathbf{y}})$. Then, the perplexity of $\hat{\mathbf{y}}$ and $\mathbf{y}$ are given by the language-specific GPT2 models (Schweter, 2020; Radford et al., 2019; Zhao et al., 2019) released on HuggingFace (Wolf et al., 2020) respectively. If the cross-entropy quality score of $(\mathbf{x}, \hat{\mathbf{y}})$ is smaller than the threshold of β_1 and the perplexity loss reduction value of $\mathbf{y} - \hat{\mathbf{y}}$ is at least β_2, then the translation $\hat{\mathbf{y}}$ is most likely better than $\mathbf{y}$. We can treat $\mathbf{y}^{\neg i:j}$ as the masked version of MT and $\mathbf{r}^{a:b}$ as the correct alternative. β_1 and β_2 are the hyper-parameters of the alignment.

Finally, we get the aligned training data ($\mathbf{x}$, $\mathbf{y}^{\neg i:j}$, $\mathbf{r}^{a:b}$) from the triplets $(\mathbf{x}, \mathbf{y}, \mathbf{r})$.

4 Experiment

4.1 Corpus and Setup

Parallel corpora for data augmentation in Section 3.1 and 3.3 and monolingual corpora for Section 3.2 are all downloaded from WMT22 general translation task[1]. For English ↔ German, WikiMatrix (Schwenk et al., 2021), News Commentary v16, Common Crawl Corpora, and Tilde MODEL Corpora (Rozis and Skadiņš, 2017) are used as parallel corpus. For English ↔ Chinese, parallel corpus we used includes UN Parallel Corpus V1.0 (Ziemski et al., 2016) and all parallel corpora from CCMT corpus (Yang et al., 2019) except for the casict2015 corpora. For monolingual corpora, News Commentary and News Crawl are used for all three languages, and Leipzig Corpora (Goldhahn et al., 2012) is also used for Chinese and German.

[1] https://statmt.org/wmt22/translation-task.html

	En-De	De-En	En-Zh	Zh-En	Average
XLM-R	25.12	27.40	32.48	21.25	26.56
Naïve Transformer	28.15	30.08	35.01	24.20	29.36
Dual-source Transformer	28.09	30.23	35.10	24.29	29.43
SA-Transformer	29.48	31.20	**36.28**	25.51	30.62
TSMind	**47.44**	**45.02**	26.41	**31.78**	**37.66**

Table 4: Sacre-BLEU on the test sets of WeTS (Yang et al., 2021)

Then the filtering strategies proposed in Section 3.1 are applied to the raw parallel data. The number of data remained after every filtering step can be found in Table 2.

We download monolingual data from WMT22, and get a total of 45.02 million German, 14.68 million English and 10.01 million Chinese monolingual sentences.

For data augmentation in Sections 3.2 and 3.3, we use the NMT models for English ↔ German and English ↔ Chinese released by Yang et al. (2021)[2] to translate the source sentences. And the hyper-parameter β_1 and β_2 to filter aligned phrases are set to 2.5 and 0.05, respectively.

4.2 Model Training

As mentioned in Section 2, a well-trained NMT model is a good starting point for the TS model. For English ↔ German, we initialize the weights with the NMT models released by Ng et al. (2019) (Winner of WMT'19). For English ↔ Chinese, the one-to-many and many-to-one mBART50 models (Tang et al., 2020) are used.

We use the fairseq toolkit (Ott et al., 2019) to train and evaluate our model. Hyper-parameters are set to the same as examples in the fairseq toolkit except that we reset the learning rate at the beginning of the first phase training and beam size is set as 6 during inference.

4.3 Experimental Results

We evaluate the TSMind by calculating the Sacre-BLEU (Post, 2018) of the top-1 generated translation suggestion candidate on the golden reference. Results of the validation sets of WMT22 are shown in Table 3. Without first-phase training, we get much worse performances. This demonstrates that a large amount of pseudo corpora contributes much to the model. However, without the second-phase training (i.e. without the human-labeled data), we cannot obtain a good translation suggestion model with only pseudo corpora either. Therefore, the design of the two-phase training and the pseudo corpora are essential to set good translation suggestions.

Since the development set of WMT'22 is not the same as the test set used in Yang et al. (2021), to make a fair comparison, we also report the Sacre-BLEU on the test set of WeTS in Table 4. Results of all baseline systems are reported by Yang et al. (2021). TSMind outperforms the strong baseline, SA-Transformer, significantly with a gap of 7.04 BLEU on average for all four language pairs. We notice that TSMind does not perform well on the English to Chinese language pair. The reason might be that the pre-trained model we use is the one-to-many model of mBART50, and the multilingual decoder is not well-trained for Chinese. For example, on the English to Chinese news translation test set of WMT'20 (Barrault et al., 2020), mBART50 only achieves a Sacre-BLEU value of 30.79, while the Sacre-BLEU of state-of-the-art is 49.2.

5 Conclusion

In this paper, we present our translation suggestion systems, TSMind, for the WMT 2022 Translation Suggestion Task. Different from previous work, we use well-trained NMT models as the pre-trained models and applied a two-phase training strategy.

We explore three data augmentation strategies from previous work and utilize the dual conditional cross-entropy model to filter out low-quality augmented data. The leader board finally shows that our submissions are ranked first in three of four language directions in the Naive TS task of WMT22 Translation Suggestion task.

References

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, et al. 2009. Statistical approaches to

[2] https://github.com/ZhenYangIACAS/WeTS

computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Lynne Bowker. 2002. *Computer-aided translation technology: A practical introduction*. University of Ottawa Press.

Lynne Bowker. 2014. Computer-aided translation: Translator training. In *Routledge encyclopedia of translation technology*, pages 126–142. Routledge.

Lynne Bowker and Des Fisher. 2010. Computer-aided translation. *Handbook of translation studies*, 1:60–65.

Rajen Chatterjee. 2019. Automatic post-editing for machine translation. *arXiv preprint arXiv:1910.08592*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Miguel Domingo, Alvaro Peris, and Francisco Casacuberta. 2016. Interactive-predictive translation based on multiple word-segments. In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, pages 282–291.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).

Jesús González-Rubio, Daniel Ortiz-Martínez, Francisco Casacuberta, and José Miguel Benedi Ruiz. 2016. Beyond prefix-based interactive translation prediction. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 198–207, Berlin, Germany. Association for Computational Linguistics.

Spence Green, Jason Chuang, Jeffrey Heer, and Christopher D Manning. 2014. Predictive translation memory: A mixed-initiative system for human language translation. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 177–187.

Spence Green, Jeffrey Heer, and Christopher D Manning. 2015. Natural language translation at the intersection of ai and hci. *Communications of the ACM*, 58(9):46–53.

Rebecca Knowles and Philipp Koehn. 2016. Neural interactive translation prediction. In *Conferences of the Association for Machine Translation in the Americas: MT Researchers' Track*, pages 107–120, Austin, TX, USA. The Association for Machine Translation in the Americas.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

Kis Balázs Lengyel, István and Gábor Ugray. 2004. Memoq: A new approach to computer-assisted translation.

Adam Lopez. 2008. Statistical machine translation. *ACM Computing Surveys (CSUR)*, 40(3):1–49.

Jun Lu, Xin Ge, Yangbin Shi, and Yuqi Zhang. 2020. Alibaba submission to the WMT20 parallel corpus filtering task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 979–984, Online. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. *arXiv preprint arXiv:1907.06616*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Álvaro Peris, Miguel Domingo, and Francisco Casacuberta. 2017. Interactive neural machine translation. *Computer Speech & Language*, 45:201–220.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Roberts Rozis and Raivis Skadiņš. 2017. Tilde MODEL - multilingual open data for EU languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 263–265, Gothenburg, Sweden. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Stefan Schweter. 2020. German gpt-2 model.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Muyun Yang, Xixin Hu, Hao Xiong, Jiayi Wang, Yiliyaer Jiaermuhamaiti, Zhongjun He, Weihua Luo, and Shujian Huang. 2019. Ccmt 2019 machine translation evaluation report. In *China Conference on Machine Translation*, pages 105–128. Springer.

Zhen Yang, Yingxue Zhang, Ernan Li, Fandong Meng, and Jie Zhou. 2021. Wets: A benchmark for translation suggestion. *arXiv preprint arXiv:2110.05151*.

Zhe Zhao, Hui Chen, Jinbin Zhang, Xin Zhao, Tao Liu, Wei Lu, Xi Chen, Haotang Deng, Qi Ju, and Xiaoyong Du. 2019. Uer: An open-source toolkit for pre-training models. *EMNLP-IJCNLP 2019*, page 241.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, Portorož, Slovenia. European Language Resources Association (ELRA).

Transn's Submissions to the WMT22 Translation Suggestion Task

Hongbao Mao **Wenbo Zhang** **Jie Cai** **Jianwei Cheng**
Transn IOL Research, Wuhan, China
{hubben.mao, albert01.zhang, jay.cai, nevil.cheng}@transn.com

Abstract

This paper describes the Transn's submissions to the WMT2022 shared task on Translation Suggestion. Our team participated on two tasks: Naive Translation Suggestion and Translation Suggestion with Hints, focusing on two language directions Zh→En and En→Zh. Apart from the golden training data provided by the shared task, we utilized synthetic corpus to fine-tune on DeltaLM (ΔLM), which is a pre-trained encoder-decoder language model. We applied two-stage training strategy on ΔLM and several effective methods to generate synthetic corpus, which contribute a lot to the results. According to the official evaluation results in terms of BLEU scores, our submissions in Naive Translation Suggestion En→Zh and Translation Suggestion with Hints (both Zh→En and En→Zh) ranked 1st, and Naive Translation Suggestion Zh→En also achieved comparable result to the best score.

1 Introduction

Combining machine translation (MT) and human translation (HT) is becoming a popular way in translation practice, which uses a typical way of post edit (PE) – the human translators are asked to provide alternatives for the incorrect word spans in the results generated by MT (Green et al., 2013; Bahdanau et al., 2015; Vaswani et al., 2017; Zouhar et al., 2021; Yang et al., 2021). In order to improve the efficiency of PE, researchers proposed translation suggestion (TS) to provide the sub-segment suggestions for the annotated incorrect word spans, and experiments show that TS can substantially reduce translators' cognitive loads and the post-editing time (Wang et al., 2020; Lee et al., 2021; Yang et al., 2021).

This paper describes the contribution of Transn IOL Research to the WMT22 Translation Suggestion shared task, where systems were submitted to two tasks: 1) Naive Translation Suggestion; 2) Translation Suggestion with Hints. For both tasks we trained the models on pre-trained encoder-decoder language model ΔLM (Ma et al., 2021) with the corpus which were synthesized deliberately, then submitted the ensemble results of the trained models. Our main contributions are:

. We utilized the pre-trained language model ΔLM to generate TS, which gets good results on the shared tasks, and much lower computational budget than training from raw Transformers (Vaswani et al., 2017; Junczys-Dowmunt and Grundkiewicz, 2018; Yang et al., 2021) as well as better quality.

. Apart from the provided golden data annotated by expert translators, we proposed the constructing methods for silver and bronze data to train TS system based on parallel corpus and the NMT models provided by the shared tasks, which contributes a lot for the final results.

. Based on the Naive Translation Suggestion models, we proposed an effective algorithm for the task of Translation Suggestion with Hints, which improves BLEU scores significantly.

The rest of this paper is organized as below. Section 2 is a brief description for Translation Suggestion shared task of WMT2022. Section 3 presents our system, including data constructing and the training process with ΔLM. Section 4 reports experimental results in the participated language directions. Finally, we conclude our work in Section 5.

2 Translation Suggestion Tasks

Translation Suggestion is a new task on WMT2022, which includes two sub-tasks.

Task 1 - Naive Translation Suggestion: This sub-task focuses on the scenario where the user

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1205 - 1210
December 7-8, 2022 ©2022 Association for Computational Linguistics

selects the incorrect span of the MT sentence without entering any information, the model outputs the alternatives automatically. Consider the source sentence x, the MT sentence m, the incorrect span selected by the user as w, the alternative y and the model parameter θ, the naive TS can be formulated as: $P(y|x, m, w, \theta)$.

Task2 - Translation Suggestion with Hints: In actual applications, users usually have general ideas of what they want. If they are dissatisfied with all the suggestions provided by Naive TS, they are willing to enter some hints for the model to generate more accurate suggestions. Given the hints h provided by users, the sub-task 2 can be formulated as: $P(y|x, m, w, h, \theta)$. In this task, we take the $top - k$ initial characters of the alternative words as the hint, and the k is randomly selected for each example.

Task 1 includes 4 language directions (En⇔Zh and En⇔De) and Task 2 includes 2 language directions (En⇔Zh). We participated En⇔Zh language directions for both tasks.

3 Implemented Systems

We fine-tune the pre-trained language model ΔLM on synthetic data and the task golden data for Task1, then adjust N-best parameter along with an optimization algorithm for Task2. The details are described in this section.

3.1 Pre-trained Model

ΔLM is a pre-trained multilingual encoder-decoder model, which outperforms various strong baselines on both natural language generation and translation tasks (Ma et al., 2021). Its encoder and decoder are initialized with the pre-trained multilingual encoder InfoXLM (Chi et al., 2020), and trained in a self-supervised way. ΔLM's pre-training tasks include span corruption on monolingual data and translation span corruption on bilingual data. We choose ΔLM as the pre-trained model for TS task because the pre-training task of translation span corruption is similar to TS. The only difference is that ΔLM masks spans in target sentence as well as spans in source sentence on bilingual data, which follows the idea from mT6 (Chi et al., 2021), but TS only masks one span in target sentence.

We use ΔLM-base model in our experiments, which has 360M parameters, 12-6 encoder-decoder layers, 768 hidden size, 12 attention heads and 3072 FFN dimension.

3.2 Construct Synthetic Data

The golden data provided by the TS tasks are annotated by expert translators, which are expensive and labor-consuming. Since the 15k golden data are far from enough to fine-tune a ΔLM model, we propose several methods to construct synthetic data for TS on parallel corpus and the specified NMT models. These synthetic data are named as silver or bronze data according to its constructing complexity as well as effect contribution.

Silver Data Silver data are constructed on parallel corpus and additional models or tools. We implemented two kinds of silver data construction.

1) The data are obtained via difference comparison on MT and target sentences. Given a parallel corpus sentence pair of source and target sentence, we first translate the source sentence by the NMT model (which is used to generate the train/dev/test data of the TS task and released to all task participants), then compute edit distance (ED) between MT and target sentence to measure the cost of editing from MT to target. We choose ED metric of LCS (Longest Common Subsequence) (Bergroth et al., 2000), which means only insertion and deletion operations are allowed (not substitution operation). ED is usually calculated by dynamic programming, and it can indicate the words which are inserted or deleted from MT to target sentence by a trace-back approach. So we can get a TS span by concatenating all words between the first and last edited words in target sentence. Table 1 shows an example for it. This can be formulated as:

$$TS = diff(NMT(source), target) \quad (1)$$

Thus, $(source, target_{diff\_mask}, TS)$ is the constructed train data, and $target_{diff\_mask}$ is the masked translation where the TS span is replaced with a placeholder. If the edited parts in target sentence is too long, it will induce a long TS span and short masked translation, so we filter out such data by a threshold.

2) The data are constructed by masking special parts on target sentences. By browsing the golden TS train data, we found there were certain regularity. Apart from the haphazard TS spans, NEs (Named Entity) and non-translated elements (especially digits) inclined to be mistranslated. So we can focus on constructing synthetic data by masking and predicting NEs and non-translated elements in target sentences. We use spaCy[1] NER function

[1] https://spacy.io/

MT	4.6.1 Suspension for Contractor reasons
$target$	4.6.1 Suspension because of Contractor reasons
$difference$	4.6.1 Suspension <add>because</add> <del>for</del><add>of</add> Contractor reasons
TS	because of
$target_{diff\_mask}$	4.6.1 Suspension <mask> Contractor reasons

Table 1: An example of synthetic data by difference comparison on MT and target sentences.

to extract such spans in target sentences, and select NE labels of PERSON, LOC, ORG, PRODUCT, MONEY and QUANTITY. This can be formulated as:

$$TS = NER(target) \tag{2}$$

Thus, $(source, target_{NER\_mask}, TS)$ is the constructed train data, and $target_{NER\_mask}$ is the masked translation where the TS span is replaced with a placeholder.

Bronze Data Bronze data are sampled directly on parallel corpus. Sampling on parallel corpus is straightforward and simple but effective for TS model. This method is also used by (Yang et al., 2021). Given the sentence pair (source,target) in the parallel corpus, we denote $target^{\backslash i:j}$ as a masked version of target sentence where its fragment from position i to j is replaced with a placeholder ($1 \leq i \leq j \leq |target|$). The $target^{i:j}$ denotes the fragment of target from position i to j. We treat $target^{i:j}$ and $target^{\backslash i:j}$ as the TS and masked translation respectively. This can be formulated as:

$$TS = target^{i:j} \tag{3}$$

So we get the constructed train data $(source, target^{\backslash i:j}, TS)$.

When the target language is Chinese, we tokenize the target sentence by Jieba[2] before sampling on it.

3.3 Training Process

We perform two-stage fine-tuning on ΔLM for training the TS models. In the first stage, we use the silver and bronze train data to fine-turn on the original ΔLM model. In the second stage, we continue to fine-tune on the results of the first stage with the golden train data. Because there are much more train data and time consumption in the first stage than that of the second stage, we just train one model for stage 1, but train several models for stage 2 with different parameters considering the plan of model ensemble. The details will be described in Section 4.

3.4 Optimization Algorithm for TS Candidates with Hints

For Task2, we use the same models as that of Task1 to generate TS candidates, and the minor adjustment is just generating more outputs with a larger N-best value during predicting. Given TS candidates by the initial predicting order, our optimization algorithm is simple and effective. Firstly, each TS candidates is converted to a string consisting of the first character of the words in TS, and secondly, we compute LCS (Bergroth et al., 2000) between each string and the hint by the candidates order, then choose the longest LCS from the results. If there are multiple longest LCSs, just choose the first one by the candidates order. Finally, the TS candidate corresponding to the longest LCS is selected as the best TS.

For Chinese language, first of all the TS candidates should be converted to phonetic symbols word by word, then perform the above process. We use pypinyin[3] to get phonetic symbols of Chinese words.

4 Experiments

We present the performance of the implemented models on the dev and test datasets, as well as some additional analysis.

4.1 Data Used

In addition to the golden train and dev data provided by the TS tasks, other data we used to train TS models are from WMT22 general translation task[4], and just part of the bilingual data are used.

Data Used for Zh→En Direction The original parallel corpus for generating synthetic data are 14 million ParaCrawl v9 Zh⇔En and 15 million UN Parallel Corpus V1.0 Zh⇔En bilingual data. Following the data constructing methods in section 3.2, all of the constructed silver and bronze data are 110 million. We sampled 4 times on different posi-

[2] https://github.com/fxsjy/jieba

[3] https://github.com/mozillazg/python-pinyin

[4] https://www.statmt.org/wmt22/translation-task.html

Dataset	Stage	Zh→En	En→Zh
dev set	Stage 1	15.62	25.10
	Stage 2	28.15	38.08
test set	Stage 2	28.42	39.71

Table 2: BLEU of two stages on dev or test sets for Zh⇔En language directions

tions for every sentence when constructing bronze data.

Data Used for En→Zh Direction Besides the original parallel corpus of Zh→En direction, we added 6 million CCMT corpus. The data constructing methods are the same as Zh→En direction, and we finally got 120 million silver and bronze data.

4.2 Results of Task 1

Following the two-stage fine-tuning process described in section 3.3, Table 2 summarizes the results of Task 1 for Zh⇔En language directions.

On stage 1, we fine-tune ΔLM-base with the constructed silver and bronze data. All models are implemented on top of the open source toolkit Fairseq[5]. We train on 6 GeForce RTX 3090 GPUs. The optimizer is Adam (Kingma and Ba, 2014) with $\beta 1 = 0.9$ and $\beta 2 = 0.98$. The learning rate is 6e-5 with a warming-up step of 8000. The models are trained with the label smoothing cross-entropy, and the smoothing ratio is 0.1. All the dropout probabilities are set to 0.3. The gradient accumulation is used due to the high GPU memory consumption, and we set max-tokens = 1600 and update-freq = 64. To speed up the training process, we conduct training with half precision floating point (FP16). We validate on dev set every 1000 updates, and the early stop patience is 5. Under these training parameters, the model converges at epoch 3.

On stage 2, we use the golden train and dev data provided by the TS tasks, and continue to fine-tune on the checkpoint with the best validation performance of stage 1. Only a few training parameters were adjusted on stage 1. The learning rate is reduced to 3e-5. In order to apply model ensemble strategies, the dropout varies in [0.1, 0.2, 0.3], and the update-freq varies in [3, 4, 5] with the fixed max-tokens 1600. The submissions are ensemble results of all models trained on stage 2.

Dataset & parameter	Zh→En	En→Zh
test set, N-best=100	39.95	48.60

Table 3: BLEU of test set with hints when N-best=100

4.3 Results of Task 2

Task 2 intends to predict more accurate translation suggestions under additional hints. So we enlarge the N-best value gradually from 5 to 100 to generate more TS candidates, and search the optimal TS by the algorithm described in section 3.4. Figure 1 shows the results on dev set where the N-best value is set in [5, 10, 20, 30, 50, 80, 100]. It seems that the BLEU rises with the increase of N-best value, but the gains diminish when N-best exceeds 50.

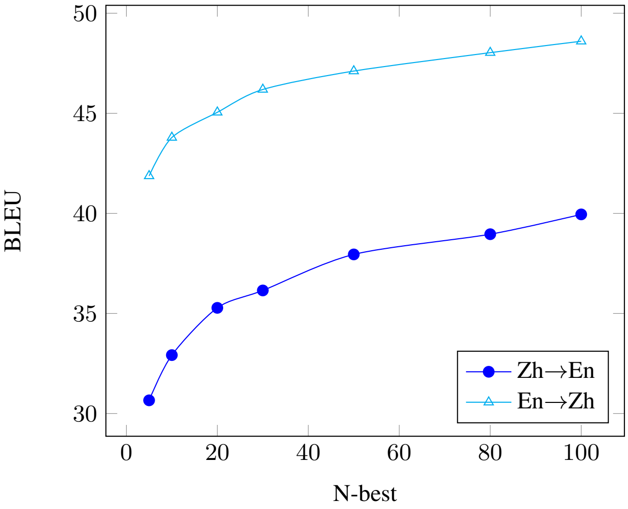


Figure 1: BLEU of dev set with hints for different N-best values

We get the final submissions on test set under N-best = 100, and the official BLEU scores are shown in Table 3 for Zh→En and En→Zh language directions.

4.4 Results Analysis Considering TS Accuracy

In practice, TS is designed to replace the incorrect span in target sentence during PE, so an absolutely accurate TS is important for post-editing translators. Therefore we analyze the accuracy indicator for TS in this section. Predicting an absolutely accurate TS relies heavily on TS length, then we analyze it based on different TS lengths, as well as top-k predictions, considering that instead of the top predicted TS, an accurate but top-k-located TS is also valuable for PE through the interactive options. Here top-k predictions are generated in the same

[5] https://github.com/facebookresearch/fairseq

TS Len		**=1**	**≤3**	**≤5**	**≤10**	**All**
dev TS, Num=2767 **(Proportion)**		**1279** **(46.2%)**	**2181** **(78.8%)**	**2488** **(89.9%)**	**2709** **(97.9%)**	**2767** **(100%)**
Top-1 predictions	Positive Num	657	900	931	949	950
	Accuracy	**51.4%**	**41.3%**	**37.4%**	**35.0%**	**34.3%**
Top-3 predictions	Positive Num	806	1159	1198	1217	1217
	Accuracy	**63.0%**	**53.1%**	**48.2%**	**44.9%**	**44.0%**
Top-5 predictions	Positive Num	943	1379	1434	1458	1458
	Accuracy	**73.7%**	**63.2%**	**57.6%**	**53.8%**	**52.7%**

Table 4: TS accuracy analysis on dev set for Zh→En language direction

way as the TS candidates in Section 3.4.

For Zh→En language direction, Table 4 shows that if there is just one word in TS, the accuracy is 51.4% for the top predictions; and the accuracy reaches 63.0% or 73.7% if we consider top-3 or top-5 predictions. Similarly, the accuracy is 41.3%, 53.1%, or 63.2% if considering top-1, top-3, or top-5 predictions respectively for the TSs no more than 3 words. The accuracy decreases gradually as the TS length increases. A significant finding is that even on the whole dev set, the accuracy still reaches 52.7% if we consider the top-5 predictions. Therefore, the accuracy indicator may help us determine when and how the TS options are activated.

4.5 Effects of Training Procedure and Synthetic Data

The two-stage fine-tuning procedure is essential for our results. If stage 1 is not applied, which means just fine-tuning ΔLM on the golden data, we get very low BLEU scores, i.e., 2.19 in Zh→En language direction on dev set. If stage 2 is not applied and just fine-tuning ΔLM on the synthetic silver and bronze data, the BLEU scores are 15.62 in Zh→En and 25.10 in En→Zh (see Table 2), with a decrease of about 13 BLEU score than the full two-stage fine-tuning procedure.

The effects of the synthetic silver and bronze data are also analyzed. Table 5 lists the results in Zh→En language direction for the single silver or bronze data on stage 1 and stage2. It shows that the silver synthetic data plays a more important role for the final performance than the bronze data.

Systems	**Zh→En**	
	Stage 1	**Stage 2**
silver & bronze data	15.62	28.15
only silver data	13.53	26.11
only bronze data	10.24	24.06

Table 5: Effects of the synthetic silver and bronze data for Zh→En language direction on dev set

5 Conclusions

We present the Transn IOL Research submissions of the WMT2022 shared task on Translation Suggestion. Our system is implemented with two-stage fine-tuning on ΔLM, which is a pre-trained encoder-decoder language model. To improve the performance, we construct synthetic data by difference comparison, named-entity masking and random sampling on parallel corpus. We propose an effective algorithm to choose optimal translation suggestion with hints. The accuracy indicator of TS is also analyzed for more efficient PE in practice. On the participated 4 tracks of En⇔Zh language directions, we achieved best scores on 3 tracks and comparable result on another track.

Effective translation suggestions benefit a lot for post editing. In the future, we plan to research field related and fine-grained TS models to improve system performance, and will integrate these advanced techniques in our translation practice.

References

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. 3rd International Conference on Learning Representations, ICLR 2015 ; Conference date: 07-05-2015 Through 09-05-2015.

L. Bergroth, H. Hakonen, and T. Raita. 2000. A survey of longest common subsequence algorithms. In *Proceedings Seventh International Symposium on String Processing and Information Retrieval. SPIRE 2000*, pages 39–48.

Zewen Chi, Li Dong, Shuming Ma, Shaohan Huang, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2021. mt6: Multilingual pretrained text-to-text transformer with translation pairs. *CoRR*, abs/2104.08692.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2020. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. *CoRR*, abs/2007.07834.

Spence Green, Jeffrey Heer, and Christopher D. Manning. 2013. The efficacy of human post-editing for language translation. In *ACM Human Factors in Computing Systems (CHI)*.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2018. Ms-uedin submission to the WMT2018 APE shared task: Dual-source transformer for automatic post-editing. *CoRR*, abs/1809.00188.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. Cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Dongjun Lee, Junhyeong Ahn, Heesoo Park, and Jaemin Jo. 2021. Intellicat: Intelligent machine translation post-editing with quality estimation and translation suggestion. *CoRR*, abs/2105.12172.

Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, and Furu Wei. 2021. Deltalm: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders. *CoRR*, abs/2106.13736.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Qian Wang, Jiajun Zhang, Lemao Liu, Guoping Huang, and Chengqing Zong. 2020. Touch editing: A flexible one-time interaction approach for translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 1–11, Suzhou, China. Association for Computational Linguistics.

Zhen Yang, Yingxue Zhang, Ernan Li, Fandong Meng, and Jie Zhou. 2021. Wets: A benchmark for translation suggestion. *arXiv preprint arXiv:2110.05151*.

Vilém Zouhar, Martin Popel, Ondřej Bojar, and Aleš Tamchyna. 2021. Neural machine translation quality and post-editing performance. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10204–10214, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Improved Data Augmentation for Translation Suggestion

Hongxiao Zhang[1], **Siyu Lai**[1], **Songming Zhang**[1], **Hui Huang**[2], **Yufeng Chen**[1]*
Jinan Xu[1] and **Jian Liu**[1]
[1]Beijing Jiaotong University, Beijing, China
[2]Harbin Institute of Technology, Harbin, China
{hongxiaozhang,siyulai,smzhang22,chenyf,jaxu,jianliu}@bjtu.edu.cn,
huanghui_hit@126.com

Abstract

Translation suggestion (TS) models are used to automatically provide alternative suggestions for incorrect spans in sentences generated by machine translation. This paper introduces the system used in our submission to the WMT'22 Translation Suggestion shared task. Our system is based on the ensemble of different translation architectures, including Transformer, SA-Transformer, and DynamicConv. We use three strategies to construct synthetic data from parallel corpora to compensate for the lack of supervised data. In addition, we introduce a multi-phase pre-training strategy, adding an additional pre-training phase with in-domain data. We rank second and third on the English-German and English-Chinese bidirectional tasks, respectively.

1 Introduction

Translation suggestion (TS) is a scheme to simplify Post-editing (PE) by automatically providing alternative suggestions for incorrect spans in machine translation outputs. Yang et al. (2021) formally define TS and build a high-quality dataset with human annotation, establishing a benchmark for TS. Based on the machine translation framework, the TS system takes the spliced source sentence $\mathbf{x}$ and the translation sentence $\tilde{\mathbf{m}}$ as the input, where the incorrect span of $\tilde{\mathbf{m}}$ is masked, and its output is the correct alternative $\mathbf{y}$ of the incorrect span. The TS task is still in the primary research stage, to spur the research on this task, WMT released the translation suggestion shared task.

This WMT'22 shared task consists of two subtasks: Naive Translation Suggestion and Translation Suggestion with Hints. We participate in the former, which publishes the bidirectional translation suggestion task for two language pairs, English-Chinese and English-German, and we participate in all language pairs.

Our TS systems are built based on several machine translation models, including Transformer (Vaswani et al., 2017), SA-Transformer (Yang et al., 2021), and DynamicConv (Wu et al., 2018). To make up for the lack of training data, we use parallel corpora to construct synthetic data, based on three strategies. Firstly, we randomly sample a sub-segment in each target sentence of the golden parallel data, mask the sampled sub-segment to simulate an incorrect span, and use the sub-segment as an alternative suggestion. Secondly, the same strategy as above is used for pseudo-parallel data with the target side substituted by machine translation results. Finally, we use a quality estimation (QE) model (Zheng et al., 2021) to estimate the translation quality of words in each translation output sentence and select the span with low confidence for masking, and then, we utilize an alignment tool to find the sub-segment corresponding to the span in the reference sentence and use it as the alternative suggestion for the span.

Considering that there is a domain difference between the synthetic corpus and the human-annotated corpus, we add an additional pre-training phase. Specifically, we train a discriminator and use it to filter sentences from the synthetic corpus that are close to the golden corpus, which we deem as in-domain data. After pre-training with large-scale synthetic data, we perform an additional pre-training with in-domain data, thereby reducing the domain gap. We will describe our system in detail in Section 3.

2 Related Work

The translation suggestion (TS) task is an important part of post-editing (PE), which combines machine translation (MT) and human translation (HT), and improves the quality of translation by correcting incorrect spans in machine translation outputs by human translators. To simplify PE, some early scholars have studied translation prediction (Green

[*] Yufeng Chen is the corresponding author.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1211 - 1216
December 7-8, 2022 ©2022 Association for Computational Linguistics

et al. (2014), Knowles and Koehn (2016)), which provides predictions for the next word (or phrase) when given a prefix. And some scholars have also studied prediction with the hints of translators (Huang et al., 2015).

In recent years, some scholars have devoted themselves to researching methods to provide suggestions to human translators. Santy et al. (2019) present a proof-of-concept interactive translation system that provides human translators with instant hints and suggestions. Lee et al. (2021) utilize two quality estimation models and a translation suggestion model to provide alternatives for specific words or phrases for correction. Yang et al. (2021) propose a transformer model based on segment-aware self-attention, provide strategies for constructing synthetic corpora, and released the human-annotated golden corpus of TS, which became a benchmark for TS tasks.

3 Method

In this section, we describe the translation suggestion system, followed by our strategies for building synthetic corpora, and finally the details of the additional pre-training phase.

3.1 Translation Suggestion System

As defined by Yang et al. (2021), given the source sentence $\mathbf{x}$, its translation sentence $\mathbf{m}$, the incorrect span $\mathbf{w}$ in $\mathbf{m}$, and its corresponding correct translation $\mathbf{y}$, the translation suggestion task first masks the incorrect span $\mathbf{w}$ in $\mathbf{m}$ to get $\mathbf{m}^{-\mathbf{w}}$, and then maximizes the following conditional probabilities:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{m}^{-\mathbf{w}}; \boldsymbol{\theta}) \quad (1)$$

where $\boldsymbol{\theta}$ is the parameters of the model.

The construction of the TS system is based on common machine translation models. We introduce the models used in our TS system below:

- **Transformer-base (Vaswani et al., 2017).** The naive transformer model. The encoding and decoding layers are both set to 6, the word embedding size is set to 512, and the attention head is set to 8.

- **Transformer-big (Vaswani et al., 2017).** The widened transformer model. The encoding and decoding layers are both set to 6, the word embedding size is set to 1024, and the attention head is set to 16.

- **SA-Transformer (Yang et al., 2021).** The segment-aware transformer model, which replaces the self-attention of the naive transformer with the segment-aware self-attention, further injects segment information into the self-attention, so that it behaves differently according to the segment information of the token. Its parameter settings are the same as those of Transformer-base.

- **DynamicConv (Wu et al., 2018).** The dynamic convolution model that predicts a different convolution kernel at every time-step. We set both encoding gated linear unit (GLU) and decoding GLU to 1 in the experiment.

3.2 Build Synthetic Corpora

Since there are few golden corpora available for training, it is necessary to build a synthetic corpus to make up for the lack of data. We build synthetic data through the following three strategies and use the mixed data for model pre-training.

3.2.1 Building on Golden Parallel Data

Following the method of Yang et al. (2021), we construct synthetic data on the large-scale golden parallel corpus. Given a sentence pair $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ and $\mathbf{r} = \{r_1, r_2, \ldots, r_m\}$ from the golden parallel corpus, we randomly sample a sub-segment $\mathbf{w} = \{r_i, r_{i+1}, \ldots, r_j\}$ of $\mathbf{r}$, we mask the sub-segment in sentence $\mathbf{r}$ to get $\mathbf{r}^{-\mathbf{w}} = \{r_1, r_2, \ldots, r_{i-1}, [\text{MASK}], r_{j+1}, \ldots, r_m\}$, and use $\mathbf{w}$ as an alternative suggestion. We perform statistics on the length of golden data to determine the length of masked spans, which is more in line with the golden distribution.

3.2.2 Building on Pseudo Parallel Data

The prediction of alternative suggestions requires the translation context, which cannot be provided by the golden parallel corpus. Therefore, we use the MT model provided by the shared task to infer the source of the large-scale parallel corpus to generate the pseudo-parallel corpus. Then we still follow Yang et al. (2021) and use the same way as described in Section 3.2.1 to construct synthetic data on the pseudo corpora consisting of source sentences and machine translation output sentences.

3.2.3 Building with Quality Estimation

The TS task is to predict the correct alternative proposal given the translation context. However, when sampling on the golden parallel corpus, the context

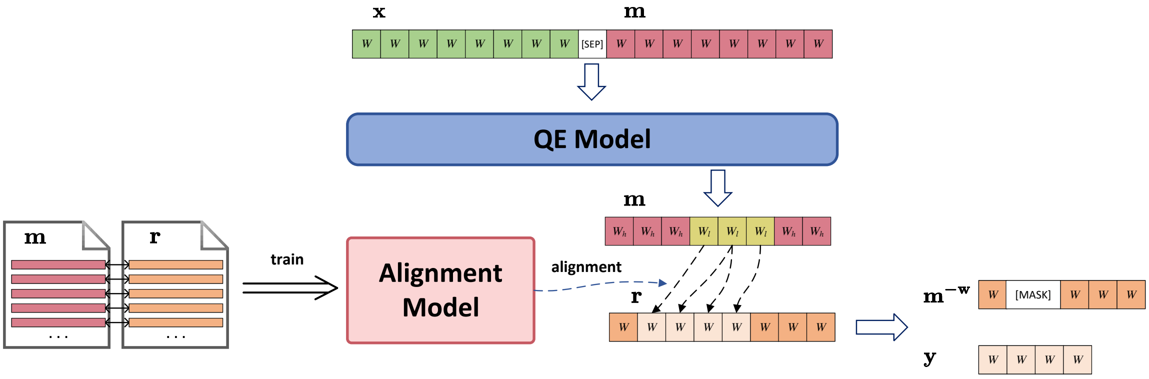


Figure 1: Schematic diagram of building synthetic corpora with quality estimation. $\mathbf{x}$ is the source sentence, $\mathbf{m}$ is the machine translation sentence, $\mathbf{r}$ is the reference sentence, and W_h and W_l represent words with high and low confidence, respectively.

does not match the translation output, and when sampling on the pseudo-parallel corpus, the alternative suggestions may be incorrect. Therefore, the above two construction strategies are not optimal.

We explore a method that is closer to the real scenarios, as shown in Figure 1. First, the word-level translation quality estimation (QE) model is used to estimate the confidence of the words in each translation sentence, and the continuous span with low confidence (that is, poor translation) is selected. Then, the translation sentence is aligned with the reference sentence through the alignment model, and the sub-segment corresponding to the span in the reference is selected as an alternative suggestion.

More specifically, we use a masked language model as our QE model, following the method of Zheng et al. (2021). To train the QE model, we splice the source sentence $\mathbf{x}_i$ and the reference sentence $\mathbf{r}_i$ of the large-scale golden parallel corpus, where some words in $\mathbf{r}_i$ are masked to get $\mathbf{r}_i^{-w}$, and the QE model is optimized to minimize the following loss function:

$$\mathcal{L} = -\sum_{i=1}^{N} \log p(\mathbf{r}_i^{w} | \mathbf{x}_i, \mathbf{r}_i^{-w}; \boldsymbol{\theta}) \qquad (2)$$

where N is the number of golden parallel sentences, $\mathbf{r}_i^{w}$ is the masked part of the reference sentence and $\boldsymbol{\theta}$ is the model parameter.

During inference, the source and translation sentences of the pseudo-parallel corpus are spliced and fed into the QE model. The model scores the word of the translation sentence according to the recovery probability of it after being masked, and words with lower scores are considered poor translations.

After that, we train a word alignment model (Lai et al., 2022) using the translated sentences and reference sentences. To ensure high alignment quality, we filter out sentences with lengths less than 5 and greater than 100 and randomly sample 5M sentence pairs for training. We use the trained alignment model to align the machine translation sentence and the reference sentence. The sub-segment in the reference that aligns with the poorly translated span described above is selected as an alternative suggestion.

3.3 Additional Pre-Training Phase with In-Domain Data

The sources of data used to construct large-scale synthetic corpus and human-annotated golden corpus are domain different. To bridge this difference, we introduce an additional pre-training stage. We filter data similar to the golden corpus as in-domain data, which are used as pre-training for the next phase after pre-training model with a large-scale synthetic corpus.

In particular, we use BERT (Devlin et al., 2019) to construct a discriminator to identify in-domain data. The discriminator consists of a binary classifier trained to distinguish between in-domain and out-of-domain sentences. The source sentences from the golden corpus as positive examples and source sentences from the synthetic corpus as negative examples are used to train this discriminator. We upsample the golden corpus by 10 times, and randomly subsample the same amount of sentences from the synthetic corpus. For each input source sentence, the discriminator predicts the probability that the sentence is in-domain. Sentences with probabilities greater than a certain threshold are

Direction	Train	Valid	Test
en⇒de	12387	1890	989
de⇒en	9308	1849	986
en⇒zh	14759	2733	1000
zh⇒en	15207	2767	1000

Table 1: The statistics of golden corpora in four translation directions.

Corpus	golden	pseudo	with QE
LS en⇔de	9.8M	9.8M	4.7M
LS en⇔zh	20M	20M	–
IND en⇒de	0.8M	0.8M	0.4M
IND de⇒en	0.7M	0.7M	0.3M

Table 2: Statistics of constructed synthetic data in our experiments, where LS stands for large-scale data and IND stands for in-domain data.

discriminated as in-domain sentences.

After the above two phases of pre-training, we use the human-annotated golden corpus for fine-tuning and test the final model.

4 Experiments and Results

4.1 Setup

We have submitted English-Chinese (en-zh) and English-German (en-de) bidirectional translation suggestion tasks. We mix en-zh data from WMT'19 and WikiMatrix, and en-de data from WMT'14 and WikiMatrix, respectively, to construct a synthetic dataset. We use the golden Train, Valid and Test set provided by this shared task, and the data statistics are shown in Table 1. We follow Yang et al. (2021) to preprocess the data, and mix the data constructed by the three strategies described in Section 3.2 as our large-scale synthetic data. The statistics of the constructed large-scale (LS) synthetic data and in-domain (IND) synthetic data are shown in Table 2. Note that for the experiments in the en-zh translation direction, we do not apply the construction strategy with QE and the pre-training phase with in-domain data. All our models are implemented based on Fairseq (Ott et al., 2019). We use the same data on each model for two phases of pre-training and fine-tuning.

System	Translation direction			
	zh-en	en-zh	de-en	en-de
Baseline	25.51	**36.28**	31.20	29.48
Ours	**28.56**	33.33	**36.30**	**42.61**

Table 3: BLEU scores on the WMT 2022 TS test set.

System	BLEU
Do nothing	18.24
+ on golden and pseudo corpus	26.91
+ with quality estimation	30.72
+ IND pre-training phase	32.95

Table 4: BLEU scores on the English-German development set for systems based on the SA-Transformer model under different strategies.

Model	BLEU
Transformer-base (A)	32.92
Transformer-big (B)	34.73
SA-Transformer (C)	32.95
DynamicConv (D)	34.03
Ensemble (A + B + C + D)	**35.81**

Table 5: BLEU scores on the development set for systems under different models in the English-German direction.

4.2 Results

We report the results of our method on the development and test set of the translation suggestion task of WMT'22. SacreBLEU[1] is used to compute the BLEU score as quality estimates relative to a human reference. We report the experimental results of our system and the baseline system (Yang et al., 2021) on the test set in Table 3, and for the baseline system, we directly use their experimental results.

As can be seen from Table 3, our system beats the baseline system in three translation directions, especially in the en-de direction, where our system surpasses the baseline by 13.13 BLEU.

We also report the results of the system on the development set of English-German translation directions to analyze the effectiveness of different models and strategies. In Table 4, we show the results of the system based on the SA-Transformer model under different strategies. "Do nothing" means we only train with the provided training set. It can be seen that the strategy of constructing synthetic data with quality estimation (QE) and the additional pre-training with the in-domain (IND) data stage can

[1] https://github.com/mjpost/sacrebleu

bring about a great improvement.

In Table 5, we present the results of systems based on different models and the model ensemble. It can be seen that in the case of the single-model system, the Transformer-big and DynamicConv models achieve better results. Besides, the ensemble model brings obvious improvement and achieves the best results.

5 Conclusion

We describe our contribution to the Translation Suggestion Shared Task of WMT'22. We propose a strategy to construct synthetic data with the quality estimation model to make the constructed data closer to the real scenarios. Furthermore, we introduce an additional phase of pre-training with in-domain data to reduce the gap between synthetic corpus and golden corpus. Experimental results demonstrate the effectiveness of our strategy. Considering the heavy labor of annotating TS data, we think data augmentation is the most important strategy that should be addressed. In the future, we will put more effort into the data generation method, to make the most of openly-accessible parallel data.

Limitations

The strategy of constructing synthetic data based on quality estimation proposed in this paper can automatically sample the incorrectly translated spans in the translations, and find the correct alternative suggestions through the alignment. It is a solution that conforms to real scenarios, and the experimental results have also proved that it is effective. However, our approach to generating synthetic data via QE still has some limitations. First, the quality estimation and alignment phases require a large additional time overhead. And second, the segments from the reference sentences may not fit into the context of the masked translation sentences due to grammar constraints. We hope to explore better solutions in future research.

Acknowledgements

The research work descried in this paper has been supported by the National Key R&D Program of China (2020AAA0108001) and the National Nature Science Foundation of China (No. 61976016, 61976015, and 61876198). The authors would like to thank the WMT'22 shared task organizers for organizing this competition and for providing open source code and models, as well as the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Spence Green, Sida I Wang, Jason Chuang, Jeffrey Heer, Sebastian Schuster, and Christopher D Manning. 2014. Human effort and machine learnability in computer aided translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1225–1236.

Guoping Huang, Jiajun Zhang, Yu Zhou, and Chengqing Zong. 2015. A new input method for human translators: integrating machine translation effectively and imperceptibly. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Rebecca Knowles and Philipp Koehn. 2016. Neural interactive translation prediction. In *Proceedings of the Association for Machine Translation in the Americas*, page 107–120.

Siyu Lai, Zhen Yang, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. 2022. Cross-align: Modeling deep cross-lingual interactions for word alignment. *arXiv preprint arXiv:2210.04141*.

Dongjun Lee, Junhyeong Ahn, Heesoo Park, and Jaemin Jo. 2021. Intellicat: Intelligent machine translation post-editing with quality estimation and translation suggestion. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 11–19.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Sebastin Santy, Sandipan Dandapat, Monojit Choudhury, and Kalika Bali. 2019. Inmt: Interactive neural machine translation prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 103–108.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. 2018. Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*.

Zhen Yang, Yingxue Zhang, Ernan Li, Fandong Meng, and Jie Zhou. 2021. Wets: A benchmark for translation suggestion. *arXiv preprint arXiv:2110.05151*.

Yuanhang Zheng, Zhixing Tan, Meng Zhang, Mieradilijiang Maimaiti, Huanbo Luan, Maosong Sun, Qun Liu, and Yang Liu. 2021. Self-supervised quality estimation for machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3322–3334.

Author Index