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 fine-tune 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 fine-tune 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.

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.

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.

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.