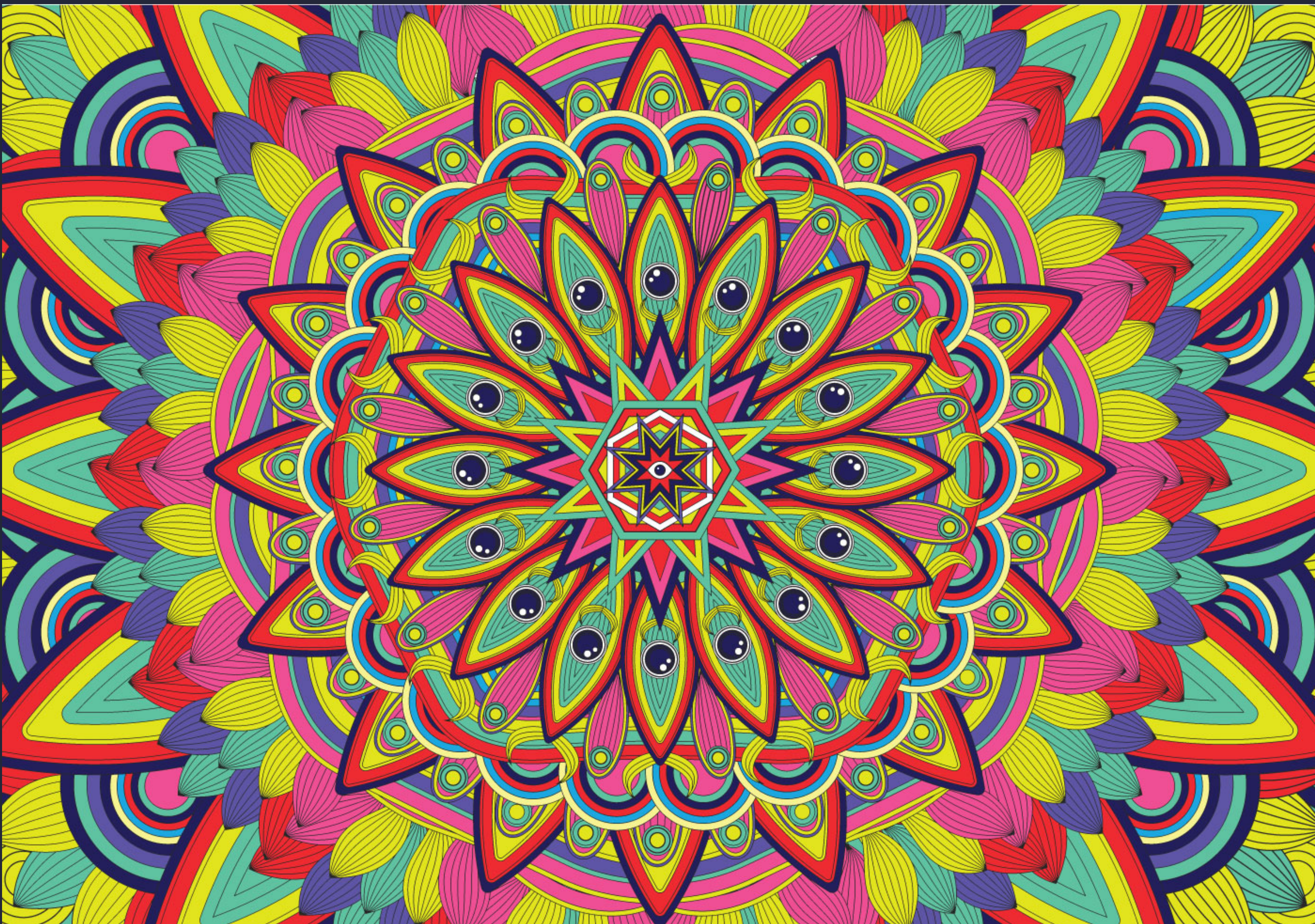
LSD 2023

Proceedings of the 2023 CLASP Conference on Learning with Small Data

Editors: Ellen Breitholtz, Shalom Lappin, Sharid Loáiciga, Nikolai Ilinykh, and Simon Dobnik



Gothenburg and online
11–12 September 2023

CLASP centre for linguistic theory and studies in probability

CLASP Papers in Computational Linguistics, Volume 5
University of Gothenburg

CLASP Conference Proceedings, Volume 2

Front-cover photo:
psychedelic pattern, https://www.freevector.com/psychedelic-pattern-28724.

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

ISSN 2002-9764
ISBN 979-8-89176-000-4

Message from the organisers

We are happy to welcome you to the CLASP Conference on Learning with Small Data (LSD 2023)! This volume consists of the papers presented at the LSD conference held at the Department of Philosophy, Linguistics and Theory of Science (FLoV), University of Gothenburg on September 11–12, 2023.

The purpose of our conference is to bring together researchers from several areas of NLP, addressing datasets, methods and limits of *effective* (machine) learning with **small data** containing natural language and associated multi-modal information. The conference covers areas such as machine learning, natural language processing, language technology, computational linguistics, theoretical linguistics, psycholinguistics, as well as artificial intelligence, cognitive science, ethics, and policy.

Current deep learning systems require large amounts of data in order to yield optimal results. Despite ever-increasing model and data size, these systems have achieved remarkable success across a wide range of tasks in NLP, and AI in general. However, these systems possess a number of limitations. Firstly, the models require a significant amount of time for pre-training, and modifying them proves to be challenging. As a result, much NLP research is shaped by what can be achieved with large transformers. This has marginalised important computational learning questions for which they are not well suited. Second, due to the substantial resources necessary for their development, they have become the preserve of technological companies. Researchers are now positioned as consumers of these systems, restricted to fine-tuning them for experimental work on downstream tasks. Thirdly, the complexity, size, and mode of computation of transformers have obscured the process through which they derive generalisations from data. This opacity has created a challenge in comprehending precisely the reasons behind their success or failure in different scenarios. Finally, comparison with human learning and representation has become increasingly difficult, given the large disparity in accessible data and learning time between transformers and humans. Therefore, the cognitive interest of deep learning has receded.

Papers were invited on topics from these and closely related areas, including (but not limited to): small-scale neural language modelling, both text and multi-modal; training corpus and test task development; visual, dialogue and multi-modal inference systems; neurolinguistic and psycho-linguistic experimental approaches to human language processing; semantics and pragmatics in neural models; dialogue modelling and linguistic interaction; formal and theoretical approaches to language production and comprehension; language acquisition in the context of computational linguistics; statistical, machine learning, reinforcement learning, and information theoretic approaches that embrace small data; methodologies and practices for annotating datasets; visual, dialogue and multi-modal generation; text generation in both the dialogue and document settings; semantics-pragmatics interface; social and ethical implications of the development and application of large or small neural language models, as well as relevant policy implications and debates.

This conference aims to initiate a discussion about these related topics and to examine various approaches and how they can mutually inform each other. The event includes 4 keynote talks, 10 peer-reviewed long papers, 5 peer-reviewed short papers, 5 peer-reviewed student papers, and 9 non-archival presentations.

We would like to thank all our contributors and programme committee members, with special thanks to CLASP for organising the hybrid conference and the Swedish Research Council for funding CLASP.

Ellen Breitholtz, Shalom Lappin, Sharid Loáiciga, Nikolai Ilinykh, and Simon Dobnik

Gothenburg

September 2023

Organising Committee

Program Chairs

Ellen Breitholtz, University of Gothenburg, Sweden
Shalom Lappin, University of Gothenburg, Sweden
Sharid Loáiciga, University of Gothenburg, Sweden

Proceedings Chairs

Simon Dobnik, University of Gothenburg, Sweden
Nikolai Ilinykh, University of Gothenburg, Sweden

Local Arrangements Organisers

Asad Sayeed, University of Gothenburg, Sweden
Susanna Myyry, University of Gothenburg, Sweden

Programme Committee

Maxime Amblard	University of Lorraine
Alexander Berman	University of Gothenburg
Raffaella Bernardi	University of Trento
Jean-Philippe Bernardy	Chalmers Technical University
Ellen Breitholtz	University of Gothenburg
Stergios Chatzikyriakidis	University of Crete
Rui Chaves	University at Buffalo
Alexander Clark	University of Gothenburg
Ariel Cohen	Ben-Gurion University of the Negev
Robin Cooper	University of Gothenburg
Philippe de Groote	Inria Nancy - Grand Est
Simon Dobnik	University of Gothenburg
Markus Egg	Humboldt University
Adam Ek	University of Gothenburg
Katrin Erk	University of Texas at Austin
Arash Eshghi	Heriot-Watt University
Chris Fox	University of Gothenburg
Jonathan Ginzburg	Université Paris-Diderot
Eleni Gregoromichelaki	University of Gothenburg
Xudong Hong	Saarland University
Julian Hough	Queen Mary University of London
Christine Howes	University of Gothenburg
Nikolai Ilinykh	University of Gothenburg
Elisabetta Jezek	University of Pavia
Richard Johansson	Chalmers Technical University
Aram Karimi	University of Gothenburg
Ruth Kempson	King’s College London
Nikhil Krishnaswamy	Colorado State University
Shalom Lappin	University of Gothenburg
Staffan Larsson	University of Gothenburg
Sharid Loáiciga	University of Gothenburg
Vladislav Maraev	University of Gothenburg
Yuval Marton	University of Washington
Elin McCready	Aoyama Gakuin University
Louise McNally	Universitat Pompeu Fabra
Gregory Mills	Kingston University
Marie-Francine Moens	KU Leuven
Joakim Nivre	Uppsala University
Bill Noble	University of Gothenburg
Manfred Pinkal	Saarland University
Massimo Poesio	Queen Mary University of London
Sameer Pradhan	cemantix.org & UPenn (Linguistic Data Consortium)
Violaine Prince	University of Montpellier
Matthew Purver	Queen Mary University of London
James Pustejovsky	Brandeis University
Chatrine Qwaider	Chalmers Technical University
Christian Retoré	University of Montpellier
Mehrnoosh Sadrzadeh	University College London

Asad Sayeed	University of Gothenburg
David Schlangen	University of Potsdam
William Schuler	The Ohio State University
Sabine Schulte im Walde	University of Stuttgart
Vidya Somashekarappa	University of Gothenburg
Tim Van de Cruys	KU Leuven
Carl Vogel	Trinity College Dublin
Alessandra Zarcone	Augsburg University of Applied Sciences
Sina Zarrieß	University of Bielefeld

Invited Speakers

Aurélie Herbelot, University of Trento
Tal Linzen, New York University & Google
Danielle Matthews, University of Sheffield
Shalom Lappin, University of Gothenburg

Invited talk: Aurélie Herbelot

Decentralised semantics

Large Language Models (LLMs) are currently the dominant paradigm in the field of Natural Language Processing. But their enormous architecture, coupled with an insatiable hunger for training data, makes them ill-suited for many purposes, ranging from fundamental linguistic research to small business applications. The main argument of this talk is that the monolithic architecture of LLMs, and by extension their reliance on big data, is a direct consequence of a lack of semantic theory in the underlying model. As an alternative, I will explore a modular architecture based on concepts from model theory, which lends itself to decentralised training over small data. Starting from research in linguistics and cognitive science, I will summarise evidence against the view that language competence should "live" in a single high-dimensional space. I will then review various computational models of meaning at the junction between formal and distributional approaches, and show how they can be combined into a modular system. Finally, I will present a possible implementation where learning takes place over individual situation types, at low dimensionality. This decentralised approach has natural benefits in terms of accessibility and energy efficiency.

Invited talk: Danielle Matthews

How children learn to use language through interaction

This talk will chart out pragmatic development with a focus on the real-world experiences that allow infants to start using language for social communication and permit children to use it at ever more complex levels. Following a working definition of pragmatics in the context of human ontogeny, we will trace the early steps of development, from a dyadic phase, through to intentional triadic communication and early word use before briefly sketching out later developments that support adult-like communication at the sentential and multi-sentential levels and in literal and non-literal ways. Evidence will be provided regarding the experiential basis of learning from the study of individual differences, from randomised controlled trials and from deaf infants growing up in families with little prior experience of deafness (and who are thus at risk of reduced access to interaction). This will provide a summary of elements from a forthcoming book: *Pragmatic Development: How children learn to use language for social communication*.

Invited talk: Tal Linzen

How much data do neural networks need for syntactic generalisation?

I will discuss work that examines the syntactic generalisation capabilities of contemporary neural network models such as transformers. When trained from scratch to perform tasks such as transforming a declarative sentence to a question, models generalise in ways that are very different from humans. Following self-supervised pre-training (word prediction), however, transformers generalise in line with syntactic structure. Robust syntactic generalisation emerges only after exposure to a very large amount of data, but even more moderate amounts of pre-training data begin to steer the models away from their linear inductive biases. Perhaps surprisingly, pre-training on simpler child-directed speech is more data-efficient than on other genres; at the same time, this bias is insufficient for a transformer to learn to form questions correctly just from the data available in child-directed speech.

Invited talk: Shalom Lappin

Assessing the Strengths and Weaknesses of Large Language Models

The transformers that drive chatbots and other AI systems constitute large language models (LLMs). These are currently the focus of a lively discussion in both the scientific literature and the popular media. This discussion ranges from hyperbolic claims that attribute general intelligence and sentience to LLMs, to the skeptical view that these devices are no more than “stochastic parrots”. In this talk I will present an overview of some of the weak arguments that have been presented against LLMs, and I will consider several more compelling criticisms of these devices. The former significantly underestimate the capacity of transformers to achieve subtle inductive inferences required for high levels of performance on complex, cognitively significant tasks. In some instances, these arguments misconstrue the nature of deep learning. The latter criticisms identify significant limitations in the way in which transformers learn and represent patterns in data. They also point out important differences between the procedures through which deep neural networks and humans acquire knowledge of natural language. It is necessary to look carefully at both sets of arguments in order to achieve a balanced assessment of the potential and the limitations of LLMs.

Table of Contents

Improving Few-Shot Learning with Multilingual Transfer and Monte Carlo Training Set Selection

Antonis Maronikolakis[△*] **Paul O'Grady**[▽] **Hinrich Schütze**[△] **Matti Lyra**[▽]
[△]Center for Information and Language Processing, LMU Munich
[△]Munich Center for Machine Learning
[▽]Zalando SE
antmarakis@cis.lmu.de

Abstract

In industry settings, machine learning is an attractive tool to automatize processes. Unfortunately, annotated and high-quality data is expensive to source. This problem is exacerbated in settings spanning multiple markets and languages. Thus, developing solutions for multilingual tasks with little available data is challenging. Few-shot learning is a compelling approach when building solutions in multilingual and low-resource settings, since the method not only requires just a few training examples to achieve high performance, but is also a technique agnostic to language. Even though the technique can be applied to multilingual settings, optimizing performance is an open question. In our work we show that leveraging higher-resource, task-specific language data can boost overall performance and we propose a method to select training examples per their average performance in a Monte Carlo simulation, resulting in a training set more conducive to learning. We demonstrate the effectiveness of our methods in fashion text reviews moderation, classifying reviews as related or unrelated to the given product. We show that our methodology boosts performance in multilingual (English, French, German) settings, increasing F1 score and significantly decreasing false positives.

1 Introduction

In real-life settings, machine learning methods are being applied to automate and improve processes, from content moderation to search query filtering. Large pretrained language models have the potential to bring further improvements, at the cost of data resources, with high quantities of labeled data required for effective training. Collecting, cleaning, annotating and analyzing data is an expensive, time-consuming and challenging task.

With recent advancements in modeling, it has been shown that large language models exhibit few- and zero-shot capabilities, able to solve tasks with little or no data (Wei et al., 2022). For example, with PET (Schick and Schütze, 2021a), models can be finetuned on a task using only a few training examples. Annotating a small number of examples is desirable and applicable in academic as well as real-life settings.

Model	F1	FPR
DistilDE	73.0%	30.7%
Set_{111}	73.4%	39.5%
Set_{MC}	74.7%	28.5%
mBERT_{all}	67.5%	16.2%
$\text{mBERT}_{all/MC}$	70.2%	25.9%

Table 1: Comparison (in the German setting) between performance of a production DistilBERT model (DistilDE) and the best-performing few-shot model (Set_{111}), as well as our models developed with our multilingual transfer learning (mBERT_{all}) and Monte Carlo (Set_{MC} and $\text{mBERT}_{all/MC}$) sampling methods. With our methods we can find a better balance between overall performance (F1 score) and the false positive rate.

An added complication in many scenarios is the need to develop solutions for multilingual settings. Annotating data, developing and evaluating models is increasingly more challenging when there are multiple languages to consider. This is exacerbated in settings where certain markets are larger and higher-resource than other markets. **Developing solutions that are scaleable to both high- and low-resource markets is challenging.**

We work in the domain of **customer text reviews on a fashion platform**. When a customer leaves a review on one of their purchased products, the review goes through moderation to verify whether it abides by the platform's code of conduct. For example, a review may be rejected because it contains offensive content or personal data. In our work, we are focusing on the task of identifying

[*]Work was performed while at Zalando SE.

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 1–10
September 11–12, 2023.

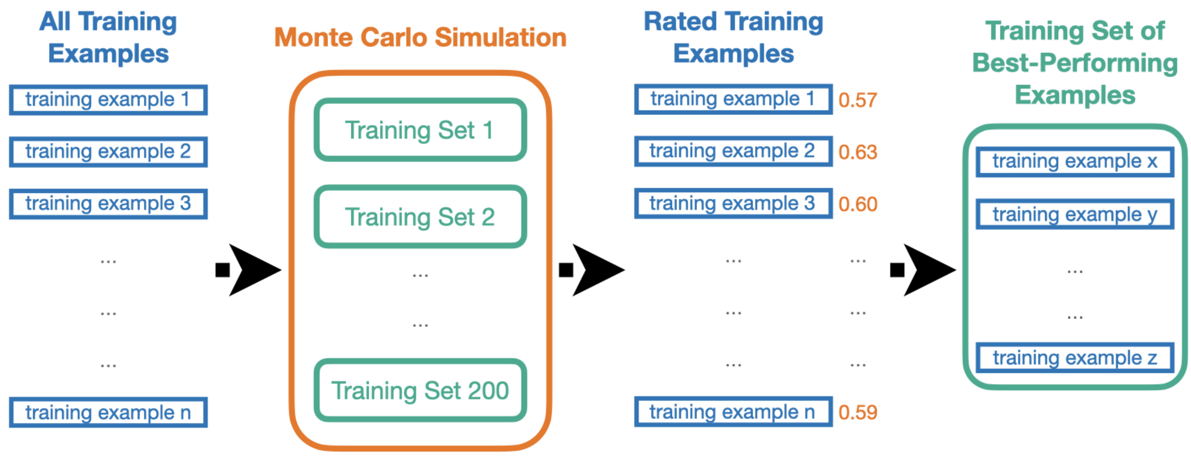


Figure 1: Overview of our proposed method. Starting from a small dataset of training examples, we perform a Monte Carlo simulation and calculate the average performance of each individual training example. Selecting the best-performing examples results in a training set more conducive to training.

whether a review is related or unrelated to the product. When a submitted review is unrelated to the product (e.g., a review that only mentions delivery time), the review is rejected.

Further, on our platform, multiple languages are covered across markets, with the majority of content written in German. Thus, **in our work, German is the focus language, with expansion to the in-domain lower-resource English and French**.

In synopsis, in an industry setting dealing with multiple markets and languages (e.g., an online shopping platform), (i) annotating large quantities of data for all languages is expensive and, (ii) the language of the dominant market makes up most of the available data. In line with these two observations, our contributions are:

1. showing that in few-shot settings, multilingual capabilities of large pretrained language models can be leveraged for better performance across languages,

2. proposing a Monte Carlo simulation method to identify training examples most conducive to learning based on a focus market (German), further improving overall performance.

We show that multilingual models finetuned on all languages perform better than their monolingual counterparts and that with our Monte Carlo selection method we can extract the training examples most conducive to learning to achieve improved performance, both in the monolingual and multilingual settings.

2 Related Work

It has been shown that large pretrained language models exhibit strong cross-lingual abilities, with cross-lingual transfer investigated extensively (Nooralahzadeh et al., 2020; K et al., 2020; Huang et al., 2019; Wu and Dredze, 2019; Pires et al., 2019; Conneau et al., 2020; Artetxe et al., 2020). In our work, we make use of cross-lingual transfer from higher- to lower-resource in-domain languages to improve performance.

With the emergent abilities of large language models (Wei et al., 2022), large models are being applied to few- and zero-shot settings (Sanh et al., 2022; Le Scao and Rush, 2021; Gao et al., 2021), showcased saliently in GPT-3, where prompting was shown to be effective across a range of tasks. To aid in few-shot learning, pattern-exploiting training (PET) was introduced in Schick and Schütze (2021a), allowing for training of large language models in few-shot settings via the use of prompts. It has been further shown that PET is competitive with models orders of magnitude larger (Schick and Schütze, 2021b).

Fu et al. (2022) showed that prompting can be employed in multilingual settings, with the authors showing that multilingual and multitask settings can be modeled without the use of language or task specific modules or training, by using prompts to leverage transfer learning capabilities. In our work we make use of prompts in a similar fashion, employing prompting to improve language transfer.

A downside with prompt-learning is the need for intricate and noisy prompt- and label-crafting, with

Language	Prompt
German	[MASK]: *r*
	r: Die Beurteilung ist [MASK]
	r ist [MASK]
English	[MASK]: *r*
	r: The review is [MASK]
	r is [MASK]
French	[MASK]: *r*
	r: L'avis est [MASK]
	r est [MASK]

Table 2: The prompts for each language, where *r* denotes the review text for each example

Language	related	unrelated
German	unabhängig	verbunden
English	related	rejected
French	pertinent	mauvais

Table 3: The labels mapping to the two classes (i.e., the verbalizer) for each language

work on the task recently gaining more traction (Lu et al., 2022; Logan IV et al., 2022; Zhao and Schütze, 2021; Schick et al., 2020; Jung et al., 2022; Mishra et al., 2022; Wu et al., 2022; Shin et al., 2020). While there has been plenty of work in prompt- and label-crafting, there has been little work in the identification of optimal training sets. While training examples can have a large impact and add significant noise during training (due to the small size of the set), selecting examples most conducive to learning is still under-explored.

3 Experimental Setup

3.1 Pattern Exploiting Training (PET)

Patter Exploiting Training (Schick and Schütze, 2021a), or PET, is a technique that reformulates examples into cloze-type questions to help task fine-tuning of language models. It has been shown to be particularly effective in low-resource settings, outperforming setups with orders of magnitude more data (Schick and Schütze, 2021b).

In our work, we employ PET during the task fine-tuning phase, training a model to predict whether a review is *related* or *unrelated* to the product. Input examples are reformulated into prompts with exactly one masked token that the model learns to fill. The model output for that masked token is mapped to one of the task classes (in our case, *related* or *unrelated*). Formally, given model vocabulary V and classes C, the verbalizer maps $C \rightarrow V$. Input x is reformulated into input $x_p \in V^*$ with exactly one masked token.

PET operates in three stages: (i) training a model for each prompt on a few annotated examples, (ii) soft-labeling a larger dataset of unlabeled data via an ensemble of prompt-trained models, (iii) training a final classifier on the soft-labeled dataset.

3.2 Models

Since PET works adjacently to Masked Language Modeling (MLM), models based on the `BERT`[1] architecture were chosen. For English, `bert-base-cased` was used. For German, we used the `bert-base-german-cased` variant. For French, we experimented with `FlauBERT` (Le et al., 2020) and `CamemBERT` (Martin et al., 2020). For `CamemBERT`, both the base and large variants were used.

As baselines we chose a logistic regression classifier and a production model for reviews moderation. The production model is based on the German `DistilBERT` model (Sanh et al., 2019). The model was originally finetuned on 20K reviews, 1271 of which were unrelated to the product and then further finetuned on 473 related and 198 unrelated reviews, to account for any shifts in data (e.g., temporal differences between original and testing data). This model is denoted with `DistilDE`.

In our work, since we have a plethora of reviews written in German, we focus on the German language for hyperparameter tuning and prompt labeling. Namely, we train using PET for 3 epochs, with a learning rate of 2e-3.

3.3 Prompt and Label Engineering

Prompts and labels were manually crafted for the German language, after empirically gauging their performance on the German development set, which is significantly larger than the English and French equivalents. We experimented both with the multi-token label variant (i.e., labels spanning multiple tokens) and multiple labels for a single

[1]Models as found on `https://huggingface.co`

class, both options were low-performing and we did not continue investigation in these directions.

Prompts for French and English were translated in a two-step process: (i) the word ‘review’ (‘Beurteilung’ in German, ‘review’ in English and ‘avis’ in French) was retrieved from the review module on the company website, which has been pre-translated by the localisation team, (ii) given the pre-translated word for ‘review’, the rest of the prompts were translated through Google Translate from German to the other two languages.

For labels, due to limitations with vanilla PET, we only chose words that span single tokens. For this reason, labels were hand-picked by the researchers to best approximate the essence of the class names, *related* and *unrelated*.

The prompts for each language are shown in Table 2. The verbalizer (label $\rightarrow$ class pairs) for each language is shown in Table 3.

3.3.1 Language Selection

In this work we focus on three languages: English, French and German. We do not verify whether customers are native speakers of these languages. Further, these languages cover multiple markets: French can be found in France, Luxembourg and Belgium, German in Austria, Germany and Switzerland, while English can be found in Germany, Ireland and the UK.

3.4 Data

Data comes from datasets of customer reviews on Zalando (an online fashion shopping platform). Reviews are submitted by customers and then moderated manually. Reviews are either accepted for publication or rejected because they do not meet the company’s policy standards. When a review is rejected, it can be rejected for one or more reject reasons. These include offensiveness, divulging of personal data, and reviews unrelated to the product. Since most rejected reviews are reviews marked as unrelated to the product, we focus on this subclass since it promises the highest return of investment. Thus, the task we are solving is the binary classification between reviews related and unrelated to the product.

Zalando’s largest market is German-speaking. Thus, (i) most reviews are written in German, (ii) there is an increased incentive to develop models to moderate German reviews. For these two reasons, we chose German as a ‘focus’ language for our work. Prompt and label engineering is conducted during experiments on the German set.

All data is made up of reviews submitted by customers after 2021 and up to June 2022, across products and product categories. For all languages, the training set (during prompt-based training) contains 8 related and 8 unrelated reviews. In English and French, the development set also contains 8 related and 8 unrelated reviews. In the German set, where more data is available, the development set is made up of 100 related and 100 unrelated reviews. For all languages, we collected 20,000 unlabeled reviews (to be soft-labeled during PET).

3.5 Monte Carlo Simulation

In few-shot settings, due to the natural scarcity of data, learning is particularly susceptible to noise in the training set. Performance relies heavily on the training set and minor, uninterpretable perturbations can affect performance drastically. Currently, selecting training examples is performed arbitrarily. We propose a method to identify which examples are most conducive to learning via a Monte Carlo simulation and selection of the examples that on average score the highest F1. The intuition behind this method is that a useful training example will on average be useful regardless of the other examples in the training set. Thus, with multiple runs, the useful training examples will on average score higher than the less useful examples.

Namely, we simulate 200 runs, via sampling 200 different training sets. For each set, we sample 16 reviews from a total of 32 possible reviews without repetition and without order-significance. The total number of combinations is intractably large. We instead sample 200 training sets due to computational considerations. Then, for each training set a model is trained using PET and evaluated on a common development set (and, finally, on the test set).

Due to the scarcity of data for English and French, the Monte Carlo simulation is performed solely on the German set. Performance was evaluated over a development set of 200 reviews (100 related and 100 unrelated to the product). The model we use is `bert-base-german-cased`.

4 Results

4.1 Monte Carlo Simulation

We performed 200 Monte Carlo runs, with `bert-base-german-cased` models trained using PET on

Method	**Dev. F1**	**Accuracy**	**F1**	**Precision**	**Recall**	**tp, fp, fn, tn**
Log. Reg.	-	72.2%	58.2%	97.5%	71.4%	1678, 44, 673, 184
DistilDE	-	88.8%	73.0%	96.8%	90.7%	2133, 70, 218, 158
Set_{113}	69.4%	**91.2%**	73.9%	95.5%	**94.8%**	**2228**, 104, **123**, 124
Set_{031}	70.3%	89.5%	71.8%	95.8%	92.5%	2175, 95, 176, 133
Set_{054}	71.0%	88.9%	70.2%	95.5%	92.2%	2168, 103, 183, 125
Set_{148}	72.7%	90.9%	74.1%	95.9%	94.1%	2212, 95, 139, 133
Set_{111}	74.9%	90.2%	73.4%	96.1%	93.1%	2188, 90, 163, 138
Set_{full}	48.6%	49.8%	43.6%	98.8%	45.3%	1068, *13*, 1283, *215*
Set_{MC}	-	89.8%	**74.7%**	97.1%	91.5%	2152, 65, 199, 163
mBERT_{de}	-	87.2%	60.1%	**92.9%**	93.0%	2186, 166, 165, 62
mBERT_{all}	-	82.3%	67.5%	**98.1%**	82.2%	1932, **37**, 419, **191**
$\text{mBERT}_{all/MC}$	-	86.0%	70.2%	97.2&	87.2%	2050, 59, 301, 169

Table 4: German performance comparison between sampled training sets performing the best on a development set versus the training set made up of the best-performing individual training examples. With bold we show the best score in each metric, except in the false positives and true negative columns, where the best performing model, Set_{full}, has not learned to recognize the positive class and thus has degenerate performance.

Method	**Accuracy**	**F1**	**Precision**	**Recall**	**tp, fp, fn, tn**
FlauBERT	60.6%	51.3%	88.1%	61.8%	1510, 204, 935, 243
CamemBERT_{base}	84.5%	45.8%	84.5%	100.0%	2445, 447, 0, 0
CamemBERT_{large}	83.9%	72.2%	92.6%	87.9%	2149, 171, 296, 276
mBERT_{fr}	**84.6%**	49.0%	84.9%	**99.3%**	**2431**, 432, **14**, 15
mBERT_{all}	77.1%	70.2%	**98.4%**	74.1%	1932, **29**, 633, **418**
$\text{mBERT}_{all/MC}$	82.5%	**74.4%**	96.5%	82.2%	2010, 72, 435, 375

Table 5: French test set performance.

sampled sets as detailed in Section 3.5. Models were evaluated on development and test sets with macro F1 score, Precision and Recall.[2] In Table 7 we see statistics on the performance of models. It is evident that performance is heavily reliant on the training set used in each iteration, with the difference between the minimum and maximum F1 scores being 40.5% in the development and 26.4% in the test set, with a standard deviation of 9.3% and 7.3% respectively. While mean performance for both sets is low (52.3% and 61.6% respectively), the maximum performance is high at 74.9% for the development and 74.1% for the test set.

We next investigate whether certain training examples are consistently more conducive to performance than other examples. In Table 8 we show the top- and bottom-3 ranked reviews based on their average F1 scores as calculated through the Monte Carlo simulation. While the worst-performing review contains multiple numbers (specifically, ‘36’ and ‘38’), which may inhibit learning, it is difficult to identify why the rest of the reviews perform better or worse. Further, reviews related and unrelated to the product are equally distributed as high- and low-performing. Nevertheless, a noticeable difference in performance can be observed, with a 4% absolute difference between the top- and bottom-scoring reviews. This exercise shows that while performance varies a lot across different training examples, it is difficult to infer why some examples perform better than others.

As a next step, we create a training set with the 16 reviews performing the best on the development set, by picking the 8 reviews related to the product with the highest score and the 8 unrelated reviews with the highest score. Combining these two sets of 8 reviews results in a 16-review training set to

[2]As implemented in https://scikit-learn.org.

Method	Accuracy	F1	Precision	Recall	tp, fp, fn, tn
BERT$_{en}$	83.2%	54.3%	98.5%	83.9%	4458, 67, 854, 101
mBERT$_{en}$	51.7%	36.1%	96.4%	52.1%	2769, 104, 2543, **64**
mBERT$_{all}$	89.7%	62.9%	**99.2%**	90.1%	4787, **39**, 525, 129
mBERT$_{all/MC}$	**93.6%**	**66.1%**	98.6%	**94.8%**	**5035**, 72, **277**, 96

Table 6: English test set performance.

Set	Min.	Max.	Mean.	Std
Dev.	34.4%	74.9%	52.3%	9.3%
Test	47.7%	74.1%	61.6%	7.3%

Table 7: Statistics of the Monte Carlo simulation performance on the development and test sets

be used in subsequent experiments. Models trained with this training set are marked with MC.

4.2 Unrelated Reviews Classification

We show that finetuning a single multilingual model on all available languages (English, French and German) outperforms its monolingual counterparts trained on each individual language. We can further improve performance by employing our proposed method of selecting training examples based on their average F1 score over a Monte Carlo simulation, improving performance in both the monolingual and multilingual settings.

For multilingual models, with mBERT$_x$ we denote the multilingual BERT model trained on the set of language x, for $x \in [fr, en, de]$. With mBERT$_{all}$ we denote the mBERT model finetuned on all languages, where the training set for German is the set that performed the best in the Monte Carlo experiments (Set$_{111}$). Finally, with mBERT$_{all/MC}$ we denote the variant of mBERT$_{all}$ where the German training set is made up of the training examples performing individually the best during the Monte Carlo experiments (the process is outlined in Section 3.5).

4.2.1 German Setting

With the greater availability of German data, this setting was chosen as the focus language of the project. While the other two languages (English and French), each have 16 reviews in the development set, there are 200 reviews available in German. For this reason, prompts and labels were crafted after evaluation on the German set (and subsequently translated into French and English) and the aforementioned Monte Carlo experiments (Section 4.1) for training example selection were performed on the German set. Results are shown in Table 4.

As baselines we compare against a logistic regression classifier trained on all 32 German reviews, as well as a production model based on German DistilBERT trained on 20K reviews.

As per the Monte Carlo experiments, 200 sets of 16 reviews were sampled from 32 total reviews, training bert-base-german-cased models on each set using PET. Here we show the five sets performing best on the development set, denoted with Set$_{xxx}$, where $x \in [0, 199]$. Set$_{MC}$ is created from the 16 best-performing training examples. For the multilingual transformer models, the training sets from English and French were used in conjunction with Set$_{111}$ (the best-performing set in the Monte Carlo experiments) forming mBERT$_{all}$, and with Set$_{MC}$ to form mBERT$_{all/MC}$.

To more fairly compare the multilingual models, which make use of more training examples (16 from each language, for 48 overall), we train a model with PET on all available German data (Set$_{full}$). This model does not seem able to generalize from all 32 examples, with very low performance when identifying reviews related to the product.

From the monolingual models, the best-performing one is Set$_{MC}$, trained on the training examples selected through the Monte Carlo simulation, outperforming the best-performing model trained during the Monte Carlo experiments (Set$_{111}$), with both a higher F1 score and lower false positive rate. Further, it outperforms the production model, DistilDE by almost 2%, despite requiring a fraction of training examples (16 versus 20K), with a slightly lower rate of false positives.

Between the multilingual models, the best-performing one is mBERT$_{all/MC}$, with an F1 score of 70.2% outperforming mBERT$_{all}$ and mBERT$_{de}$ with F1 scores 67.5% and 60.1% respectively. However, mBERT$_{all}$ has the lowest rate of false positives, with only 37 false positives versus 59 false

ID	Review	Avg. F1	Label
1	Sehr bequem und richtig süss 🥰 ich liebe baby rosa 😍🦋	63.9%	related
2	es ist ganz schade	63.9%	unrelated
3	Gutes Material auf der Haut, sitzt nicht ganz teilliert, etwas lockeres, hat es trotzdem behalten.	63.9%	related
4	Es würde ein anderes t-shirt schicken und zur nächsten würde ich wieder eine Retour machen.	60.0%	unrelated
5	Kann ich leider noch keine Bewertung abgegeben, ist noch bei Hermes wie gesagt	59.7%	unrelated
6	Ich musste eine Nummer größere bestellen, trage gewöhnlich 36. Ich hätte größe 38 kaufen sollen.	59.7%	related

Table 8: The top- and bottom-3 ranked reviews on average F1 performance from the Monte Carlo experiments for German (reviews here have been edited to preserve privacy and abide by GDPR laws).

positives from mBERT$_{all/MC}$. Nevertheless, the increase in the F1 score is significant at 2.7%, which shows that utilizing our proposed method of Monte Carlo selection and multilingual transfer performs the best for the multilingual setting too.

4.2.2 French Setting

For our experiments in French, we are comparing three monolingual models (FlauBERT, CamemBERT$_{base}$ and CamemBERT$_{large}$) with three multilingual models (i) mBERT$_{fr}$, (ii) mBERT$_{all}$, and (iii) mBERT$_{all/MC}$. Results are shown in Table 5.

Out of the three monolingual models, only CamemBERT$_{large}$ performs well, with FlauBERT having a low recall and CamemBERT$_{base}$ unable to identify reviews unrelated to the product. While CamemBERT$_{large}$ has a high F1 score, it suffers from a high rate of false positives, with 171 unrelated reviews classified as related. With our method this issue is mitigated, reducing the number of false positives to 29 and 72 with mBERT$_{all}$ and mBERT$_{all/MC}$ respectively, while at the same time keeping overall performance competitive or even better that the monolingual counterparts. While mBERT$_{all}$ performs slightly worse than CamemBERT$_{large}$, with an F1 score of 70.2% versus 72.2%, mBERT$_{all/MC}$ outperforms the monolingual model with an F1 score of 74.4%.

We can thus conclude that in French, performance improves both in reducing the false positive rate and in increasing the F1 score via multilingual transfer from the better-performing German training set selected through the Monte Carlo simulation (Set$_{MC}$) to the French set.

4.2.3 English Setting

For our experiments in English, we are comparing the monolingual BERT$_{en}$ model with three multilingual models (i) mBERT$_{en}$, (ii) mBERT$_{all}$, and (iii) mBERT$_{all/MC}$. Results are shown in Table 6.

While training mBERT solely on English does not perform well, with a very low recall score, mBERT$_{all}$ and especially mBERT$_{all/MC}$ perform well, showing that multilingual transfer and our proposed Monte Carlo selection method jointly improve performance. Namely, while the monolingual model has an F1 score of 54.3%, mBERT$_{all/MC}$ has an F1 score of 66.1%.

Unfortunately, in this case our proposed method does not provide improvements for the false positives rate. In fact, mBERT$_{all/MC}$ introduces 5 more false positives than the monolingual baseline. This could potentially be noise in the evaluation set, considering that English-language data was scarce, with only 168 unrelated reviews. Nevertheless, our proposed method improves the F1 score by 11.8% over the monolingual model.

4.2.4 Multilingual Setting

Finally, we compare in greater detail monolingual with multilingual model performance.

For our experiments in the multilingual setting, we compare three types of models. With BERT$_{mono}$ we denote the BERT model pretrained and finetuned on the corresponding language performing the best in each language setting (e.g., CamemBERT$_{large}$ for French), with mBERT$_{all}$ we denote the model trained on all language training sets and with mBERT$_{all/MC}$ we denote the multilingual model variant where the

Model	French	English	German
BERT_{mono}	72.2%	54.3%	**73.4%**
mBERT_{all}	70.2%	62.9%	67.5%
$\text{mBERT}_{all/MC}$	**74.4%**	**66.1%**	70.2%

Table 9: Comparison of monolingual and multilingual F1 scores per language on each test set.

Model	French	English	German
BERT_{mono}	38.3%	39.9%	39.5%
mBERT_{all}	**10.7%**	**23.2%**	**16.2%**
$\text{mBERT}_{all/MC}$	25.9%	42.9%	25.9%

Table 10: Comparison of monolingual and multilingual false positive rate per language on each test set.

German Set_{MC} was used instead of Set_{111}.

The monolingual models are the models that performed the best in their respective language settings on the development sets (although for French and English the development sets contain only 16 reviews). The multilingual models were all finetuned using English labels and prompts, since English data is the most prominent in mBERT's pretraining set and intuitively the best hub across languages (even though Anastasopoulos and Neubig (2020) showed that choosing English is not always the best hub-language for bilingual models, hub-language selection is out of scope for our work).

In Table 9 we compare F1 scores. In French and English, multilingual models perform best, with an average increase to the F1 score of 2% for mBERT_{all} and 11% for $\text{mBERT}_{all/MC}$. In German, while $\text{mBERT}_{all/MC}$ performs competitively, the monolingual model still performs the best. This is to be expected, considering that German was the focus language of our experiments and the German monolingual models received the majority of attention during the development stage, with hyperparameter tuning and extensive prompt engineering. On the other hand, no tuning or engineering was performed on the multilingual models.

In Table 10, we compare false positive rates (i.e., unrelated reviews that were classified as related) between monolingual and multilingual models. In many real-world settings, false positives are particularly insidious, since users are exposed to content they should not be seeing. For example, exposing customers to harmful content or publishing reviews where customers have inadvertently revealed personal information (e.g., their address or email address) is harmful. In our case, customers are exposed to information that is not related to the product, which may add confusion and affect customer trust negatively.

For this important metric, we see that the monolingual models perform badly, with the false positive rate just below 40% across all languages. This is performance that would deem such models untrustworthy for production. On the other hand, the multilingual models perform better overall. In particular, mBERT_{all} performs the best in all languages and by large margins (at least 16%). $\text{mBERT}_{all/MC}$ also performs better than the monolingual models in French and German, sporting improvements of at least 12%. In English, however, $\text{mBERT}_{all/MC}$ performs competitively but still worse than the monolingual model by 3%. An explanation for this is that in English the test set contains only a few unrelated (i.e., negative) reviews, numbering at 168. In comparison, in the German set there are 228 unrelated reviews and in the French set 447. Thus, due to the small size, it is challenging to make inference on solely unrelated reviews.

5 Conclusion

In our work, we investigate how to improve discrimination between customer reviews related and unrelated to the product in low-resource settings for English, French and German. We show that via multilingual transfer learning we can improve performance of models in English and French, leveraging the in-domain higher-resource German data, while at the same time reducing the rate of false positives across all languages.

Selecting training examples most conducive to performance in few-shot learning is of paramount importance and still an open question. We propose a method to extract such examples through a Monte Carlo simulation, selecting training examples with the highest average performance across experiments. We show that with our method we can improve performance both in monolingual and multilingual settings, outperforming baselines with orders of magnitude more data as well as all models trained on randomly-sampled sets, consistently increasing F1 scores and decreasing false positives.

Acknowledgements. The first author performed work as an intern at Zalando SE (Berlin). The first author is currently funded by the European Research Council (grant #740516).

6 Limitations

In our Monte Carlo simulation, due to computational restrictions, we only performed 200 random runs. Even though the training set produced via our selection method outperforms other methods, performance could be potentially improved even further with more runs.

Future work should focus on expanding the set of languages investigated. Due to limited data resources, we were not able to procure enough data in other languages. It is important to include not only more, but also more linguistically diverse languages in the study.

References

Antonios Anastasopoulos and Graham Neubig. 2020. Should all cross-lingual embeddings speak English? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8658–8679, Online. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. Association for Computational Linguistics.

Jinlan Fu, See-Kiong Ng, and Pengfei Liu. 2022. Polyglot prompt: Multilingual multitask promptraining.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. 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 3816–3830, Online. Association for Computational Linguistics.

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019. Unicoder: A universal language encoder by pre-training with multiple cross-lingual 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 2485–2494, Hong Kong, China. Association for Computational Linguistics.

Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. Maieutic prompting: Logically consistent reasoning with recursive explanations.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual BERT: an empirical study. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2020. Flaubert: Unsupervised language model pre-training for french. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France. European Language Resources Association.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online. Association for Computational Linguistics.

Robert Logan IV, Ivana Balazevic, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2022. Cutting down on prompts and parameters: Simple few-shot learning with language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2824–2835, Dublin, Ireland. Association for Computational Linguistics.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022. Reframing instructional prompts to GPTk's language. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 589–612, Dublin, Ireland. Association for Computational Linguistics.

Farhad Nooralahzadeh, Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein. 2020. Zero-shot cross-lingual transfer with meta learning. In *Proceedings of the 2020 Conference on Empirical Methods* *in Natural Language Processing (EMNLP)*, pages 4547–4562, Online. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5569–5578, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

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.

Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA. Association for Computing Machinery.

Mengjie Zhao and Hinrich Schütze. 2021. Discrete and soft prompting for multilingual models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8547–8555, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Smooth Sailing: Improving Active Learning for Pre-trained Language Models with Representation Smoothness Analysis

Josip Jukić Jan Šnajder
University of Zagreb, Faculty of Electrical Engineering and Computing
Text Analysis and Knowledge Engineering Lab, Croatia
{josip.jukic, jan.snajder}@fer.hr

Abstract

Developed to alleviate prohibitive labeling costs, active learning (AL) methods aim to reduce label complexity in supervised learning. While recent work has demonstrated the benefit of using AL in combination with large pre-trained language models (PLMs), it has often overlooked the practical challenges that hinder the effectiveness of AL. We address these challenges by leveraging representation smoothness analysis to ensure AL is *feasible*, that is, both effective and practicable. Firstly, we propose an early stopping technique that does not require a validation set – often unavailable in realistic AL conditions – and observe significant improvements over random sampling across multiple datasets and AL methods. Further, we find that task adaptation improves AL, whereas standard short fine-tuning in AL does not provide improvements over random sampling. Our work demonstrates the usefulness of representation smoothness analysis for AL and introduces an AL stopping criterion that reduces label complexity.[1]

1 Introduction

The notorious data hungriness of deep learning models emphasizes the importance of efficient and effective label acquisition. However, the labeling process is often tedious and expensive, ultimately slowing the development of labeled datasets and resulting in subpar models. Evolved out of a practical necessity, *active learning* (**AL**; Cohn et al., 1996; Settles, 2009) is a special family of machine learning algorithms designed to reduce *label complexity* – the number of labels that a learning algorithm requires to achieve a given performance (Dasgupta, 2011) – and thus minimize labeling costs. An AL method aims to select the most informative examples, which can be particularly useful when unlabeled data are abundant, but the labeling is costly or requires substantial expertise.

The striking success of deep learning has motivated the use of traditional AL techniques for training deep neural networks (DNNs) and the development of novel AL methods suited specifically to DNNs. In natural language processing (NLP), AL has been shown to outperform a random selection of examples in many NLP tasks (Zhang et al., 2017; Siddhant and Lipton, 2018; Ikhwantri et al., 2018). Before the widespread adoption of large pre-trained language models (PLMs), a typical AL approach to training deep models was to use task-specific neural models trained from scratch in each AL step (Kasai et al., 2019; Prabhu et al., 2019). Since PLMs fine-tuned to downstream tasks outperform standard neural models, PLMs have supplanted most of them, and researchers have begun to investigate the feasibility of AL for PLMs (Ein-Dor et al., 2020; Schröder et al., 2022). Recent work in AL experimented with several training regimes, such as PLM adaptation and specific fine-tuning techniques (Yuan et al., 2020; Margatina et al., 2022). In particular, task-adaptive pre-training (**TAPT**) has emerged as a cost-effective method for performance improvement complementary to AL (Howard and Ruder, 2018). TAPT uses additional pre-training on the unlabeled training set via masked language modeling and self-supervision. In theory, combining AL with adapted PLMs should produce greater reductions in label complexity than either of the methods in isolation. However, since research on combining AL with PLMs is still in its infancy, whether it can work consistently better than random selection in realistic conditions remains an open question.

One of the challenges in combining AL and PLMs is that, although AL is conceptually simple and promises efficiency gains, there are a host of practical challenges in deploying it in realistic conditions (Attenberg and Provost, 2011; Lowell

[1]Our code is available at https://github.com/josipjukic/al-playground

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 11–24
September 11–12, 2023.

et al., 2019). The situation is further aggravated by the fact that most AL research overlooks these challenges and resorts to unrealistic evaluation setups and resources. One of the most pervasive problems stems from using a hold-out set during training (e.g., a validation set for regularization by early stopping). In real applications, hold-out sets are unlikely to be available, as building them would require additional labeling effort the AL is meant to reduce in the first place. Another major problem is the flawed evaluation of AL methods: typically, an AL method is compared against random selection as the baseline, but the two training regimes are not kept identical, which confounds the measured effect of AL. In addition to the above-mentioned problems, there is the important practical question of when to stop the acquisition of labels, i.e., how to define the AL *stopping criterion*.

AL methods rely highly on the *acquisition model* (the underlying model used for selecting examples). Therefore, it is important to maintain good generalization properties of the acquisition model, which can be analyzed using representation smoothness. Recently, functional space theory has emerged as a valuable tool for analyzing generalization properties and expressivity of DNNs (Yarotsky, 2017; Suzuki, 2019). In particular, the *Besov space*, a general function space that can capture spatial inhomogeneity, appears convenient for such analyses (Suzuki and Nitanda, 2021).

In this work, we address the practical challenges of AL. First, we systematically evaluate the **feasibility**, where we consider an AL method to be feasible if it is both *practicable* (achievable in realistic conditions) and *effective* (consistently outperforms random selection). Concretely, we explore different learning regimes in AL on various NLP classification tasks without a validation set that is unavailable in most real-world labeling campaigns. Motivated by the effectiveness of TAPT for PLMs (Gururangan et al., 2020), we explore how TAPT combines with AL in the low-resource setup. Secondly, we leverage the representation smoothness of PLM layers in the Besov space to improve AL effectiveness. In particular, we develop *Besov early stopping*, an early stopping regularization technique that does not require a validation set, and we show that it consistently improves the model performance and reduces the variance of results for all AL methods we consider. Moreover, Besov early stopping shows promise as a surrogate for a validation set in zero- and few-shot setups for regular training without AL. We also utilize representation smoothness to develop a stopping criterion based on the smoothness of AL samples to minimize label complexity. Our experiments show a reduction in label complexity for PLMs across five NLP datasets and five AL methods. In addition, building on the idea that representation smoothness is relevant for AL, we complement our experiments with a novel AL method based on the norm of representation gradients. Both the proposed method and the existing AL methods consistently outperform random selection on PLMs with TAPT, which supports the recent findings that the training regime is more important than the choice of the AL method (Margatina et al., 2022).

Our contributions can be summarized as follows: (1) we conduct a systematic evaluation of AL methods for large PLMs and show that AL is feasible, i.e., it consistently outperforms random selection under realistic conditions, (2) we analyze the smoothness of the representation space of PLMs in AL and propose an early stopping technique that improves AL performance and stabilizes the results, (3) we discover patterns in the representation smoothness of AL samples, which we use for an effective AL stopping criterion, and (4) we introduce a representation-based AL method, competitive with other state-of-the-art AL strategies. Our results demonstrate that AL with PLMs is feasible. Even more importantly, the results indicate that representation smoothness analysis can be leveraged to improve model training in general and the effectiveness of AL in particular, opening new avenues for further research.

2 Related Work

Our work builds on several strands of research, including practical challenges in AL, combining AL with PLMs, and different training setups for AL acquisition models.

Practical challenges in AL. Despite the success of AL for many NLP tasks, studies have identified a number of practical challenges hindering the broader deployment of AL (Attenberg and Provost, 2011; Lowell et al., 2019). The most obvious problem is the unavailability of a labeled validation set, an essential resource in model training typically used for hyperparameter optimization and regularization via early stopping. Moreover, in realistic AL conditions, a labeled test set is also unavailable, making a held-out evaluation of the underlying model's quality impossible. Previous work mostly used model confidence or training error stability to evaluate the acquisition model and derive an AL stopping criterion based on that estimation (Vlachos, 2008; Bloodgood and Vijay-Shanker, 2009; Zhu et al., 2010; Ishibashi and Hino, 2021). However, these criteria have not been widely adopted as they often require tuning for specific datasets and tasks. We mitigate this by developing a task-agnostic AL stopping criterion that detects the points of the largest reduction in label complexity compared to random selection.

AL with PLMs. Only recently have large PLMs been coupled with AL. Early work concentrated mainly on the Transformer architecture (Vaswani et al., 2017) utilizing a simple training setup. More concretely, the predominant approach was to use a standard fine-tuning technique with a fixed number of training epochs, fine-tuning the model from scratch in each AL step (Ein-Dor et al., 2020; Margatina et al., 2021; Shelmanov et al., 2021; Karamcheti et al., 2021; Schröder et al., 2022). However, Mosbach et al. (2021) and Zhang et al. (2021) showed that fine-tuning in low-resource setups (scenarios with little training data) tends to be very unstable, especially when training for only a few epochs. This instability poses a serious issue, as AL often implies a low-resource setting. Moreover, fine-tuning is often sensitive to weight initialization and data ordering (Dodge et al., 2020). This instability of PLM fine-tuning also makes the AL results unstable. We address the instability issue by proposing an early stopping technique without a validation set, and we show that combining PLMs with AL is feasible.

AL training regimes. AL research took a turn from standard fine-tuning of pre-trained models to explore different training regimes and how to use them in combination with AL methods. For example, Grießhaber et al. (2020) explored how to efficiently fine-tune Transformers with AL by freezing the network's layers. Similarly, Yuan et al. (2020) explored self-supervised language modeling to estimate example informativeness for cold-start active learning. Motivated by the general success of TAPT (Gururangan et al., 2020), Margatina et al. (2022) showed that AL outperformed random sampling for PLMs with TAPT, albeit using a validation set. Similarly, Yu et al. (2022) developed a self-training approach for active learning with the addition of weighted clustering. While some training regimes seem promising for AL, the outstanding question is which regimes can consistently outperform random selection. Furthermore, considering what resources are realistically available during training, the primary concern is whether we can apply these training regimes in realistic conditions.

3 Representation in Besov Space

Due to their remarkable flexibility and adaptivity, deep learning models have gained significant traction. To explain these phenomena, researchers have leveraged function space theory to develop approximation and estimation error analysis (Yarotsky, 2017; Suzuki, 2019). Our work relies on a particular type of analysis based on the theory of Besov spaces.

3.1 Besov space

It has been shown that the expressive power of DNNs can be analyzed by specifying the target function's property such as **smoothness** (Petersen and Voigtländer, 2018; Imaizumi and Fukumizu, 2019), i.e., the number of orders of continuous derivatives it has over some domain. *Besov space* has proven to be especially convenient for such analyses, as it allows spatially inhomogeneous smoothness with spikes and jumps, which we often encounter in high-dimensional deep learning. In Besov spaces, the approximation error (expressivity)[2] and estimation error (generalizability)[3] depend on the properties of the representation space (Suzuki and Nitanda, 2021). Given these theoretical connections, representation space analysis can steer toward better generalization properties.

3.2 Besov smoothness index

We briefly describe the mathematical apparatus of the Besov space analysis, adopted with slight modifications from (Suzuki, 2019; Suzuki and Nitanda, 2021). Let $\Omega \in \mathbb{R}^d$ be a domain of functions. For a function $f : \Omega \to \mathbb{R}$ with a defined p-norm in L_p (space of measurable functions with finite p-norm) and seminorm $|f|$ defined by $x \mapsto |f(x)|$, we define $\|f\|_p := \|f\|_{L^p(\Omega)} := (\int_\Omega |f|^p dx)^{\frac{1}{p}}$

[2]The approximation error refers to the distance between the target function and the closest neural network function of a given architecture.

[3]Estimation error refers to the distance between the ideal network function and an estimated network function.

for $0 < p < \infty$. For $p = \infty$, we define $\|f\|_\infty := \|f\|_{L^\infty(\Omega)} := \sup_{x\in\Omega} |f(x)|$.

Definition 1 (Smoothness modulus)**.** *For a function $f \in L^p(\Omega)$, $p \in (0, \infty]$, $t \in (0, \infty)$, $h \in \mathbb{R}^d$, and $r \in \mathbb{N}$, the r-th modulus of smoothness of f is defined by*

$$w_{r,p}(f, t) := \sup_{\|h\|_2 \le t} \|\Delta_h^r(f)\|_p,$$

where $\Delta_h^r(f)$ is the forward difference operator of the r-th order defined as $\Delta_h^r(f)(x) := \sum_{i=0}^{r} \binom{r}{i} (-1)^{r-i} f(x + ih)$ for $[x, x + rh] \in \Omega$, and 0 otherwise.

Definition 2 (Besov space ($\mathcal{B}_{p,q}^{\alpha}$))**.** *For $0 < p, q \le \infty$, $\alpha > 0$, $r := \lfloor \alpha \rfloor + 1$, let the seminorm of the Besov space $\mathcal{B}_{p,q}^{\alpha}$ be*

$$|f|_{\mathcal{B}_{p,q}^{\alpha}} := \left(\int_0^\infty \left(t^{-\alpha} w_{r,p}(f, t)\right)^q \frac{dt}{t} \right)^{\frac{1}{q}} \tag{1}$$

*for $q < \infty$. Let $|f|_{\mathcal{B}_{p,q}^{\alpha}} = \sup_{t>0} t^{-\alpha} w_{r,p}(f, t)$ for $q = \infty$. **Besov smoothness index** of f is determined as the maximum index α for which the Besov seminorm is finite.*

Intuitively, the Besov smoothness index (*Besov smoothness* for short) quantifies the properties of DNN's representation space. More specifically, a higher index indicates higher smoothness. Because the calculation of Besov smoothness (more precisely, the integral in (1)) is intractable, we have to rely on approximations. Elisha and Dekel (2016, 2017) proposed wavelet decomposition of a random forest (RF) for approximating Besov smoothness. Wavelet decomposition of the RF establishes an order of importance of the RF nodes, while RF uses the embedded representations of an arbitrary DNN as features. For classification problems, we can normalize the inputs to $[0, 1]$ and transform the class labels into vectors in the $\mathbb{R}^{L-1}$ space by assigning each label to a vertex of a standard simplex, where L is the number of classes. This gives us the k-th layer of a neural network as a function $f_k : [0, 1]^d \to \mathbb{R}^{L-1}$. For a random forest consisting of J estimators, Elisha and Dekel (2017) proceeded by approximating the errors of each estimator $\mathcal{T}_j$ with M most important wavelets. The error function (with $r = 1$, $p = 2$) is estimated as $\sigma_M \sim c_k M^{-\alpha_k}$. Numerically, we can use an approximation $\log(\sigma_m) \sim \log(c_k) - \alpha \log(m)$, $m = 1, \ldots, M$, and find c_k and α_k through least squares, where α_k is the estimate for the Besov smoothness of f_k, i.e., the k-th layer of a DNN.

3.3 Representation smoothness

Analyzing the Besov smoothness of DNNs can unveil their representation geometry. DNNs should benefit from smoother representations, as they help the model avoid overfitting. Intuitively, "well-learned" representations will exhibit high Besov smoothness. When we decompose a PLM into wavelets sorted by relevance and use the Besov smoothness approximation described in Section 3.2, smoother representations achieve lower generalization errors with fewer wavelets.

Another relevant phenomenon for representation smoothness analysis is that the individual layers of DNNs specialize in different features. In particular, earlier layers tend to learn *generalization* features, while the deeper layers are more prone to *memorization* (Stephenson et al., 2021; Baldock et al., 2021). Following these insights, we propose using Besov smoothness to inspect the generalization properties of PLMs through the prism of layer-wise representation geometry. We hypothesize PLMs should benefit more from smoother representations in earlier layers, and we propose methods to enforce learning such representations during training.

4 Preliminaries

In this section, we describe our experimental setup, detailing the datasets, models, AL methods, and evaluation metrics.

4.1 Datasets

We select three different single-text classification tasks commonly used in the AL literature. The datasets vary in size, number of classes, and complexity, allowing for a nuanced study of AL methods. To extend our analysis to similar datasets with different levels of complexity, we also add binary versions of the multi-class tasks. In total, we work with five datasets (cf. Appendix, Table 3): (1) the question type classification dataset (**TREC-6**; Li and Roth, 2002); (2) the corresponding binary version **TREC-2** with only the two most frequent classes (*Entity* and *Human*); (3) the subjectivity dataset SUBJ of Pang and Lee (2004), which classifies the movie snippets as subjective or objective and is often used in AL benchmarks; (4) the AG's News classification dataset AGN-4 of Zhang et al. (2015); and (5) its binary version, **AGN-2**, often used in the AL literature, with two categories (*World* and *Sports*) out of four.

4.2 Models

We focus on large PLMs and include two representatives of the Transformer family, each using a different pre-training paradigm. Specifically, we experiment with BERT (Devlin et al., 2018), which uses a generative pre-training approach via masked language modeling, and ELECTRA (Clark et al., 2020), which relies on discriminative training to detect corrupted tokens induced by a small generator network. For both models, we leverage their widely used *base* variants from the Hugging Face library (Wolf et al., 2020), which consist of 12 layers.

4.3 Active learning methods

We consider six sampling strategies, including random selection, which serves as a baseline. The other five strategies are AL methods from different families.

Random selection (**RND**) selects instances uniformly from the unlabeled pool.

Maximum entropy (**ENT**; Lewis and Gale, 1994) comes from the family of *uncertainty* strategies. The method queries instances where the model is least certain, according to the criterion of maximum entropy of the prediction output.

Monte Carlo dropout (**MC**; Gal and Ghahramani, 2016) is similar to ENTROPY, but relies on the stochasticity of forward passes with dropout layers (Srivastava et al., 2014) to estimate the entropy for a given instance.

Core-set (**CS**; Sener and Savarese, 2018) promotes instance diversity by leveraging the learned representations of the acquisition model. The method aims to minimize the distance between an example in the unlabeled set and its most similar counterpart in the labeled subset.

Discriminative active learning (**DAL**; Gissin and Shalev-Shwartz, 2019) frames active learning as a classification of whether a particular instance is labeled or not to make the labeled and unlabeled sets indistinguishable. Specifically, DAL queries instances that are most likely to be in the unlabeled subset according to a trained classifier.

Representation gradients (**RG**) is a novel AL strategy we propose in this work. Similar to methods from (Huang et al., 2016; Ash et al., 2019), RG selects instances based on gradient information from the representation space. However, unlike other gradient-based methods, RG is much less computationally demanding and, therefore, suitable for resource-limited studies and realistic conditions. The method computes the mean representation gradient with respect to the embedded inputs and selects the instances with the largest gradient norm. Formally, with $\bar{\mathbf{h}}$ as the mean representation, the RG's selection criterion is $\mathrm{argmax}_{\mathbf{x}\in\mathcal{U}} \left\|\partial_{\mathbf{x}}\bar{\mathbf{h}}\right\|_2$, where $\mathcal{U}$ denotes the unlabeled set. The intuition behind RG is that the locally sharp instances in the representation space of the underlying model, i.e., the ones with large gradient norms, surprise the model the most and thus will contribute the most to a reduction in label complexity.

In our experiments, we select 50 new examples in each step of each AL experiment, using 100 examples for the warm start (randomly sampled labeled data to kick-start the model). We set the labeling budget to $1,000$ instances for easier datasets (TREC-2, AGN-2, and SUBJ) and $2,000$ instances for harder datasets (TREC-6 and AGN-4).

4.4 Evaluation

To evaluate the entire AL process, we use the area under the performance curve (**AUC**). Each step corresponds to the classification performance in terms of the F_1 score of a model trained with a certain number of labeled examples. We advocate using AUC complementary to the AL curves, as we believe it is a good approximation of AL feasibility as a summary numeric score. Since we use different training regimes in our experiments, we compare each AL strategy to random selection within the same training regime to isolate the effects of AL. Additionally, we introduce a metric to measure the direct practical gains of AL by estimating the reduction in label complexity of AL compared to random selection. For a given AL step, we compute the number of additional labels required to achieve the same performance with random selection, thus estimating the number of labels one saves when using AL. We refer to this metric as *label complexity reduction* (**LCR**).

5 Improving Active Learning

In this section, we first look into the representation geometry of PLMs by means of representation smoothness analysis. Then, we link our findings to devise a smoothness-based early stopping technique that does not require a validation set. We explore the effects of our method in different training regimes and provide a systematic evaluation of AL for PLMs in the low-resource setup.

5.1 Representation smoothness analysis

We empirically test the characteristics of Besov smoothness of PLMs. In particular, we compare the representation smoothness of PLMs in three different training regimes: (1) **short training** (ST), where models were trained for 5 epochs, (2) **extended training** (ET), where models were trained for 15 epochs, and (3) model adaptation with TAPT (cf. Appendix A.5 for details) followed by an extended training for 15 epochs (ETA). We computed the smoothness of PLM layers during training, averaged across AL steps. In each AL step, we fine-tuned the model anew.

Performance-wise, ETA yields better results than ET and ST (Table 1). Moreover, AL in the ST regime does not yield improvement over random sampling. Figure 1 shows the layer-wise smoothness for the three mentioned regimes with the addition of the overfitting regime, where we purposefully overfitted the acquisition model in each AL step by training the model for 100 epochs. In the ST regime, we observe a monotonic increase in smoothness as we progress through layers, while the smoothness in ET peaks before the last few layers. The shift of the smoothness peak is even more pronounced for TAPT with extended training. In overfitted models, we observe a flat distribution of smoothness across layers. We observe that better performance and effective AL come with a shift in smoothness distribution towards earlier layers, as displayed in ET and ETA regimes. We hypothesize that, in the low-resource setup, the deeper layers exhibit higher smoothness in the ST regime because they are prone to *heuristic memorization* – DNN relies on spurious artifacts (shortcuts) that are correlated with a target label (Bansal et al., 2022) – which may cause the model to perform poorly.

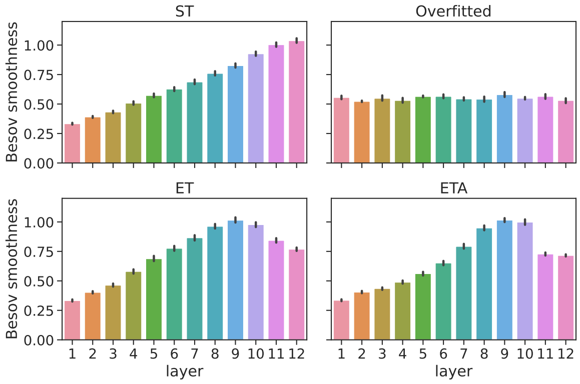


Figure 1: Besov smoothness of PLM layers for different training regimes. The scores are normalized (between 0 and 1 per layer) and averaged across datasets, models, and AL methods. The black error bars represent the standard deviation. We note that the deviation is small, indicating similar behavior across different datasets, models, and AL methods.

5.2 Besov early stopping

In the AL loop, the effect of selecting an acquisition model with poor generalization properties propagates through the AL steps. To ensure the effectiveness of AL, regularization by early stopping is often used to pre-empt overfitting in order to retain good generalization properties. However, since a validation set is often unavailable in realistic conditions, using it for early stopping renders AL impracticable. However, feasible AL needs to be both effective and practicable.

The above empirical findings on smoothness distribution across the PLM layers for the different training regimes motivate an early stopping heuristic based on representation smoothness without a validation set. We propose **BEAST** (**B**esov **ea**rly **st**opping), where we proceed with the training as long as the Besov smoothness distribution skews toward earlier layers. We define the stopping point as the epoch where the distribution skewness[4] fails to increase, i.e., when the peak of the representation smoothness ([CLS] token) fails to shift towards earlier layers for two consecutive epochs. We revert the model to the last epoch where this effect is preserved. In this way, we stop the training before the smoothness distribution flattens out, which we observe in overfitted models. We experiment with two more training regimes: $\text{ET}^{\mathcal{B}}$ and $\text{ETA}^{\mathcal{B}}$, which are just ET and ETA with BEAST.

We compare BEAST to the approaches without early stopping, where we chose the models from the last epoch. Our experiments show the difference in AL performance across different training

[4]We compute the layer-wise smoothness skewness as the Fisher-Pearson coefficient of skewness.

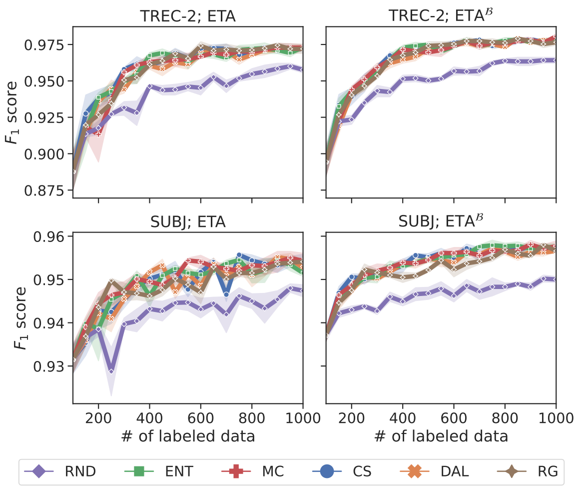


Figure 2: Active learning performance curves for BERT in terms of F_1 score. Random sampling (purple rhombs) serves as a baseline. For the sake of space, we show the results on a subset of datasets for BERT and regimes ETA and ETA$^{\mathcal{B}}$ as we obtained similar results for other configurations (cf. Figure 4 in Appendix). The results are averaged over five runs. The confidence intervals represent the standard deviation. Best viewed on a computer screen.

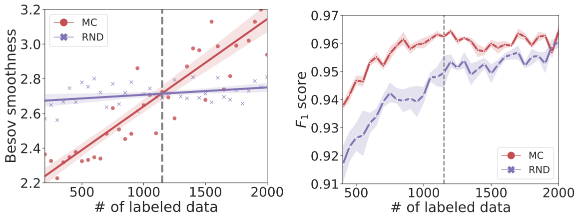
regimes. Figure 2 shows the trend of AL curves through the steps, and Table 1 provides more comprehensive comparisons with AUC as the aggregated measure of AL effectiveness. We can observe that AL coupled with ST performs poorly, and AL fails to outperform random sampling (sometimes even worse than random sampling). The ET regime generally improves performance, with AL sometimes outperforming random selection. ETA and ETA$^{\mathcal{B}}$ further improve performance over random sampling for every AL method on every dataset we used. For BERT, the difference between AL and random sampling is statistically significant in 22 out of 25 cases with ETA and in all 25 cases with ETA$^{\mathcal{B}}$. More importantly, ET$^{\mathcal{B}}$ and ETA$^{\mathcal{B}}$ outperform their counterparts without BEAST and reduce the variance of the results (cf. Appendix, Table 6). We support the hypothesis that the choice of the AL method is not as important as the training regime, as we achieve similar results for every method when AL outperforms random selection. TAPT works across the board, improving AL performance on all five datasets. With the addition BEAST, we achieve feasible AL, making it both practicable and effective. On top of that, even with random sampling, BEAST consistently yields higher scores than the model from the last epoch, showing benefits even for regular fine-tuning without AL.

		RND	ENT	MC	CS	DAL	RG
TREC-2	ST	.875	.873	.883	.881	.889	.879
	ET	.912	.932†	.934†	.929	.931	.931
	ET$^{\mathcal{B}}$	.925	.942†	.942†	.939†	.940†	.938
	ETA	.941	.959†	.957†	.960†	.957†	.958†
	ET$^{\mathcal{B}}$	.949	**.966**†	.965†	.965†	.964†	.965†
SUBJ	ST	.896	.892	.885	.901	.898	.892
	ET	.920	.922	.922	.925	.925	.920
	ET$^{\mathcal{B}}$	.928	.931	.932	.932	.933	.930
	ETA	.942	.949†	.950†	.949†	.949†	.948
	ET$^{\mathcal{B}}$	.946	**.954**†	**.954**†	**954**†	.953†	.952†
AGN-2	ST	.923	.942†	.941†	.922	.941†	.942†
	ET	.960	.969	.970†	.965	.967	.969
	ET$^{\mathcal{B}}$	.967	.974†	.975†	.972	.974†	.975†
	ETA	.974	.981†	.980	.981†	.980	.980†
	ET$^{\mathcal{B}}$	.977	**.983**†	**.983**†	**.983**†	.982†	.982†
TREC-6	ST	.706	.743†	.749†	.666	.689	.693
	ET	.867	.878	.881†	.867	.878	.867
	ET$^{\mathcal{B}}$	.873	.885	.890†	.873	.882	.875
	ETA	.909	.933†	.931†	.931†	.934†	.930†
	ET$^{\mathcal{B}}$	.925	.939†	.937†	.936†	**.940**†	.935†
AGN-4	ST	.837	.828	.824	.801	.834	.829
	ET	.869	.869	.871	.871	.880†	.875
	ET$^{\mathcal{B}}$	.875	.877	.878	.879	.886†	.881
	ETA	.891	.905†	.905†	.902†	.906†	.899†
	ET$^{\mathcal{B}}$	.894†	.908†	.908†	.905†	**.909**†	.903†

Table 1: AUC scores for random sampling and different AL methods across datasets and training regimes for BERT (cf. Appendix, Table 5). The results are averaged over 5 runs with different seeds. **Bold** numbers indicate the best AUC for each dataset. The "†" indicates when the mean AUC of an AL method is significantly different from random sampling (two-sided Man-Whitney U test with $p < .05$, adjusted for family-wise error rate with the Holm-Bonferroni method).

6 Active Sample Smoothness

In Section 5, we analyzed the Besov smoothness of layer representations of PLMs. In this section, we take a step further and examine the smoothness at the instance level. Instead of using the representations on the training set, we computed the Besov smoothness as the average across layers on the unseen (selected but not yet trained on) *active sample*, acquired by the AL method. In contrast to the seen training examples, we argue that the Besov smoothness on unseen examples can be interpreted as the amount of information the model could obtain from that sample. More precisely, the lower the smoothness of an active sample, the more informative it is for the model. In contrast, smooth samples are already well-represented and thus not as resource-effective as their less smooth counterparts.



(a) Sample smoothness (b) AL curve

Figure 3: The relationship between an active sample and random sample smoothness. Figure 3a shows how the smoothness of samples retrieved by AL (red) relates to the smoothness of random samples (violet) with fitted regression lines. The smoothness values are calculated as the average across layers. Figure 3b shows the corresponding AL performance curve. The gray dotted line indicates the intersection of active and random sample smoothness, which signals the beginning of diminishing returns of AL. We show the results for BERT in the ETA$^{\mathcal{B}}$ regime on the TREC-6 dataset as an illustration. We observe very similar patterns in other datasets. Since all of the AL methods display similar behavior in ETA$^{\mathcal{B}}$ regime, we show only MC to avoid clutter (cf. Figure 5 in Appendix for other datasets). Best viewed on a computer screen.

We compare the Besov smoothness of actively acquired samples against random samples. We consistently observe two patterns, showcased by Figure 3. First, the smoothness of random samples is uniform throughout the AL steps. The second pattern occurs in the trend of AL sample smoothness. In the early AL steps, AL sample smoothness is low, indicating sharp representations that require smoothing (by learning). As the AL procedure progresses, the acquisition model improves, and the active samples' smoothness increases. We interpret this as the model slowly consuming the information from the data pool, eventually reaching a state of "information depletion", i.e., a state in which the remaining unlabeled data provides no additional value to the model.

Stopping criterion. Our preliminary analysis of the relationship between the active and random sample smoothness motivates a simple stopping criterion, which we refer to as ALSBI (**a**ctive **l**earning **s**topping by **B**esov **i**ndex). ALSBI aims to detect when AL methods reach information depletion. We terminate the AL process when sample smoothness surpasses the average smoothness of a random sample in two consecutive steps. We disregard the first AL step, as it often takes several steps for the acquisition models to stabilize. Since we cannot compute the smoothness of a random sample (as we query only AL samples) in realistic conditions, we estimate the random sample smoothness on the warm start examples via bootstrapping. This approximation proved stable for 100 examples as the smoothness of a random sample remains stable throughout AL steps. We take the average smoothness of $1,000$ bootstrapped samples of size 50. Table 2 shows that ALSBI yields larger LCR than what one would get on average across AL steps, which supports our preliminary analysis. RG achieves the highest LCR among the tested AL methods, which we believe is due to its compatibleness to ALSBI as both the AL method and the stopping criterion are based on representation smoothness.

		ENT	MC	CS	DAL	RG
ETA	avg	.316	.312	.351	.275	.319
	ALSBI	.447	.401	.511	.282	**.521**
ETA$^{\mathcal{B}}$	avg	.385	.385	.394	.354	.382
	ALSBI	.586	.532	.488	.447	**.645**

Table 2: Average LCR across datasets and models. The scores indicate the proportion of the dataset that needs to be labeled for random sampling to match the performance of the corresponding AL method. ALSBI is compared to an average LCR throughout the AL steps (avg). The results are averaged over 5 runs. Numbers in **bold** indicate the largest LCR for a certain training regime.

7 Conclusion

In our paper, we leverage representation smoothness analysis to improve the effectiveness of active learning (AL). In realistic conditions, we show that AL with pre-trained language models (PLMs) is effective when combined with task adaptation, while standard short fine-tuning often fails. We address the problem of unavailable resources (labeled holdout sets) by developing the **B**esov **ea**rly **st**opping technique (BEAST) that does not require a validation set. For AL to be feasible, it must be both effective and practicable. BEAST meets both feasibility requirements: it improves AL performance over random sampling and reduces the variance of the performance scores across AL steps (effectiveness) while not requiring additional labeled data (practicability). Moreover, BEAST improves the performance of PLMs even in standard fine-tuning without AL, which makes it potentially useful in zero-shot and few-shot setups where a validation set could also be unavailable. We further show the usefulness of representation smoothness analysis for AL by devising a simple and effective AL stopping criterion. We corroborate the hypothesis from previous research in that the effectiveness of AL is influenced more by the training regime rather than the AL method itself. We believe that the relationship between PLMs' generalization properties, label complexity, and representation smoothness is an exciting avenue for AL, and we hope our results will motivate further research in that direction.

Limitations

To fully comprehend the significance of our findings, it is necessary to consider the limitations of this study. Firstly, we evaluate only two Transformer-based models on a small number of text classification tasks. Although we used the models with different pre-training paradigms, it is possible that the findings do not generalize across models within the same family. In addition, we used the base variants of BERT and ELECTRA, which both feature 12 layers. Since our early stopping criterion is influenced by the number of layers whose smoothness we approximate, there is a possibility that smoothness would distribute differently for models with more or fewer layers. Another limitation is that we did not investigate these models' performance on tasks other than text classification, and the results may not be generalizable to different types of NLP tasks.

Since there are many different ways to measure the quality of an AL stopping criterion and we only wanted to illustrate the usefulness of smoothness patterns, we only compared the proposed ALSBI method against an average baseline. However, a more comprehensive comparison with other approaches from the literature would provide a better understanding of the merit of our method.

Lastly, we only scratched the surface of different training regimes for PLMs in the context of AL. Many new training regimes are emerging in the field, especially the ones focused on efficiency and modularity. We leave the exploration of these methods for future work.

References

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2019. Deep batch active learning by diverse, uncertain gradient lower bounds. *CoRR*, abs/1906.03671.

Josh Attenberg and Foster Provost. 2011. Inactive learning? Difficulties employing active learning in practice. *ACM SIGKDD Explorations Newsletter*, 12(2):36–41.

Robert Baldock, Hartmut Maennel, and Behnam Neyshabur. 2021. Deep learning through the lens of example difficulty. In *Advances in Neural Information Processing Systems*, volume 34, pages 10876–10889. Curran Associates, Inc.

Rachit Bansal, Danish Pruthi, and Yonatan Belinkov. 2022. Measures of information reflect memorization patterns. In *Advances in Neural Information Processing Systems*.

Michael Bloodgood and K. Vijay-Shanker. 2009. A method for stopping active learning based on stabilizing predictions and the need for user-adjustable stopping. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 39–47, Boulder, Colorado. Association for Computational Linguistics.

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*.

David A Cohn, Zoubin Ghahramani, and Michael I Jordan. 1996. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145.

Sanjoy Dasgupta. 2011. Two faces of active learning. *Theoretical Computer Science*, 412(19):1767–1781. Algorithmic Learning Theory (ALT 2009).

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.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *CoRR*, abs/2002.06305.

Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online. Association for Computational Linguistics.

Oren Elisha and Shai Dekel. 2016. Wavelet decompositions of random forests - smoothness analysis, sparse approximation and applications. *Journal of Machine Learning Research*, 17(198):1–38.

Oren Elisha and Shai Dekel. 2017. Function space analysis of deep learning representation layers. *CoRR*, abs/1710.03263.

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.

Daniel Gissin and Shai Shalev-Shwartz. 2019. Discriminative active learning. *CoRR*, abs/1907.06347.

Daniel Grießhaber, Johannes Maucher, and Ngoc Thang Vu. 2020. Fine-tuning BERT for low-resource natural language understanding via active learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1158–1171, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Jiaji Huang, Rewon Child, Vinay Rao, Hairong Liu, Sanjeev Satheesh, and Adam Coates. 2016. Active learning for speech recognition: the power of gradients.

Fariz Ikhwantri, Samuel Louvan, Kemal Kurniawan, Bagas Abisena, Valdi Rachman, Alfan Farizki Wicaksono, and Rahmad Mahendra. 2018. Multi-task active learning for neural semantic role labeling on low resource conversational corpus. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, pages 43–50, Melbourne. Association for Computational Linguistics.

Masaaki Imaizumi and Kenji Fukumizu. 2019. Deep neural networks learn non-smooth functions effectively. In *AISTATS*.

Hideaki Ishibashi and Hideitsu Hino. 2021. Stopping criterion for active learning based on error stability.

Siddharth Karamcheti, Ranjay Krishna, Li Fei-Fei, and Christopher Manning. 2021. Mind your outliers! investigating the negative impact of outliers on active learning for visual question answering. 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 7265–7281, Online. Association for Computational Linguistics.

Jungo Kasai, Kun Qian, Sairam Gurajada, Yunyao Li, and Lucian Popa. 2019. Low-resource deep entity resolution with transfer and active learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5851–5861, Florence, Italy. Association for Computational Linguistics.

David D Lewis and William A Gale. 1994. A sequential algorithm for training text classifiers. In *SIGIR'94*, pages 3–12. Springer.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

David Lowell, Zachary C. Lipton, and Byron C. Wallace. 2019. Practical obstacles to deploying active learning. 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 21–30, Hong Kong, China. Association for Computational Linguistics.

Katerina Margatina, Loic Barrault, and Nikolaos Aletras. 2022. On the importance of effectively adapting pretrained language models for active learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 825–836, Dublin, Ireland. Association for Computational Linguistics.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 650–663, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning {bert}: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

Philipp Christian Petersen and Felix Voigtländer. 2018. Optimal approximation of piecewise smooth functions using deep relu neural networks. *Neural networks : the official journal of the International Neural Network Society*, 108:296–330.

Ameya Prabhu, Charles Dognin, and Maneesh Singh. 2019. Sampling bias in deep active classification: An empirical study. 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 4058–4068, Hong Kong, China. Association for Computational Linguistics.

Christopher Schröder, Andreas Niekler, and Martin Potthast. 2022. Revisiting uncertainty-based query strategies for active learning with transformers. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2194–2203, Dublin, Ireland. Association for Computational Linguistics.

Ozan Sener and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*.

Burr Settles. 2009. Active learning literature survey. Computer sciences technical report.

Artem Shelmanov, Dmitri Puzyrev, Lyubov Kupriyanova, Denis Belyakov, Daniil Larionov, Nikita Khromov, Olga Kozlova, Ekaterina Artemova, Dmitry V. Dylov, and Alexander Panchenko. 2021. Active learning for sequence tagging with deep pre-trained models and Bayesian uncertainty estimates. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1698–1712, Online. Association for Computational Linguistics.

Aditya Siddhant and Zachary C. Lipton. 2018. Deep Bayesian active learning for natural language processing: Results of a large-scale empirical study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2904–2909, Brussels, Belgium. 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. *Journal of Machine Learning Research*, 15(56):1929–1958.

Cory Stephenson, Suchismita Padhy, Abhinav Ganesh, Yue Hui, Hanlin Tang, and SueYeon Chung. 2021. On the geometry of generalization and memorization in deep neural networks. In *International Conference on Learning Representations*.

Taiji Suzuki. 2019. Adaptivity of deep reLU network for learning in besov and mixed smooth besov spaces: optimal rate and curse of dimensionality. In *International Conference on Learning Representations*.

Taiji Suzuki and Atsushi Nitanda. 2021. Deep learning is adaptive to intrinsic dimensionality of model smoothness in anisotropic Besov space. In *Advances in Neural Information Processing Systems*, volume 34, pages 3609–3621. Curran Associates, Inc.

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.

Andreas Vlachos. 2008. A stopping criterion for active learning. *Computer Speech & Language*, 22(3):295–312.

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.

Dmitry Yarotsky. 2017. Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114.

Yue Yu, Lingkai Kong, Jieyu Zhang, Rongzhi Zhang, and Chao Zhang. 2022. AcTune: Uncertainty-based active self-training for active fine-tuning of pretrained language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1422–1436, Seattle, United States. Association for Computational Linguistics.

Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-start active learning through self-supervised language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7935–7948, Online. Association for Computational Linguistics.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2021. Revisiting few-sample {bert} fine-tuning. In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Ye Zhang, Matthew Lease, and Byron Wallace. 2017. Active discriminative text representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Jingbo Zhu, Huizhen Wang, Eduard Hovy, and Matthew Ma. 2010. Confidence-based stopping criteria for active learning for data annotation. *ACM Transactions on Speech and Language Processing (TSLP)*, 6(3):1–24.

	TRAIN	VAL	TEST	TOTAL
TREC-2	$1,987$	159	486	$2,632$
SUBJ	$7,000$	$1,000$	$2,000$	$10,000$
AGN-2	$20,000$	$2,600$	$5,000$	$27,600$
TREC-6	$4,881$	452	500	$5,833$
AGN-4	$20,000$	$7,600$	$7,600$	$35,200$

Table 3: Dataset sizes by splits. Although we do not use a validation set (VAL) in our experiments, we report its size for completeness. We uniformly subsampled the AGN-2 and AGN-4 datasets for shorter computation time.

	BERT	ELECTRA
TREC-2	32.4	31.1
SUBJ	40.8	39.2
AGN-2	71.4	70.3
TREC-6	68.4	67.1
AGN-4	82.3	75.7

Table 4: Experiment duration in minutes for both models across datasets. We report the average runtime over five different runs and six different sampling methods (five AL methods and random sampling).

A Reproducibility

A.1 Dataset statistics

We report the sizes of the datasets per split in Table 3. The datasets contain mainly texts in English.

A.2 Models

We used base and uncased variants of the Transformer models. Specifically, we used "bert-base-uncased" for BERT and "google/electra-base-discriminator" for ELECTRA. Both models have $109,514,298$ trainable parameters each.

A.3 AL methods

MC We use ten inference cycles to approximate the entropy of the output via Monte-Carlo dropout sampling.

CS We use the [CLS] token representation from the Transformer's penultimate layer. We opt for the greedy method described in the original paper (Sener and Savarese, 2018).

A.4 Preprocessing

We use the same preprocessing pipeline on all datasets for both BERT and ELECTRA. We lowercase the tokens, remove non-alphanumeric tokens and truncate the sequence to 200 tokens.

A.5 Hyperparameters

We used a fixed learning rate of 2×10^{-5} for both models. Additionally, we set the gradient clipping to 1 during training. In the ST regime, we trained the model for 5 epochs and 15 in ET, and ETA. For TAPT, we used masked language modeling with 15% of randomly masked tokens and trained the model via self-supervision for 50 epochs with the learning rate set to 10^{-5}.

A.6 Computing infrastructure

We conducted our experiments on $4\times$ *AMD Ryzen Threadripper 3970X 32-Core Processors* and $4\times$ *NVIDIA GeForce RTX 3090* GPUs with 24GB of RAM. We used *PyTorch* version 1.9.0 and CUDA 11.4.

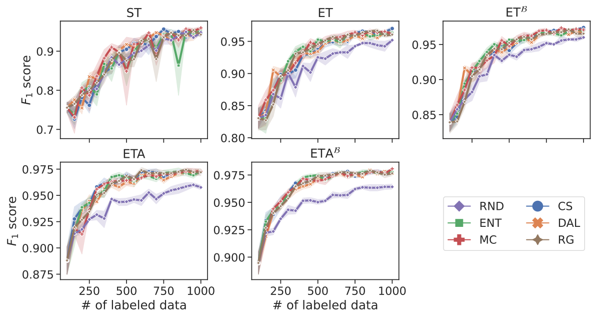
A.7 Average runtime

We report the average runtime of experiments in Table 4. We ran six sampling methods on five datasets for two models and for five different training regimes. Additionally, we repeated each experiment five times with different seeds ($[1, 2, 3, 4, 5]$). In each experiment, we re-train the model 20 times on TREC-2, SUBJ, and AGN-2 up to $1,000$ instances (20 batches of 50 instances), and 40 times on TREC-6 and AGN-4 up to $2,000$ instances (40 batches of size 50). In total, we ran 300 AL experiments.

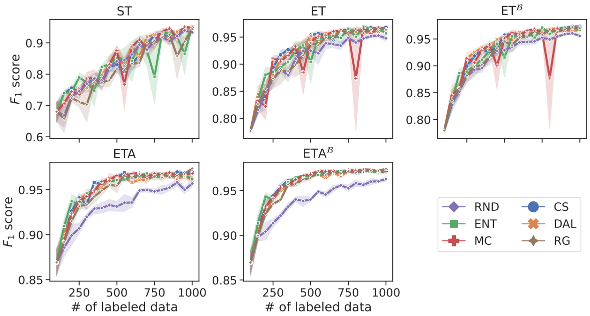
B Experiments

We report the experiments that were omitted from the main part of the paper due to space constraints. Figure 4 shows the active learning performance curves across the used datasets and for both models (BERT and ELECTRA). For the ETA and ETA$^{\mathcal{B}}$ training regimes, we observe a consistent improvement in performance compared to random sampling. We report the results for ELECTRA in Table 5, where we observed similar patterns as with BERT (cf. Table 1 in the main part of the paper). On top of that, our early stopping method reduces the variance of the results compared to other training regimes, as shown in Table 6.

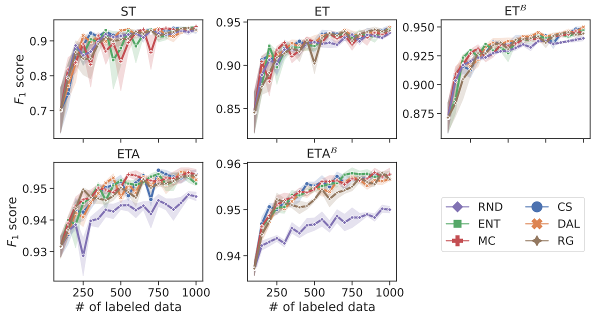
Figure 5 shows the relationship between the Besov smoothness of random and active samples. We report the smoothness of samples for each dataset. We observe a similar pattern, with a rising smoothness of actively acquired samples.



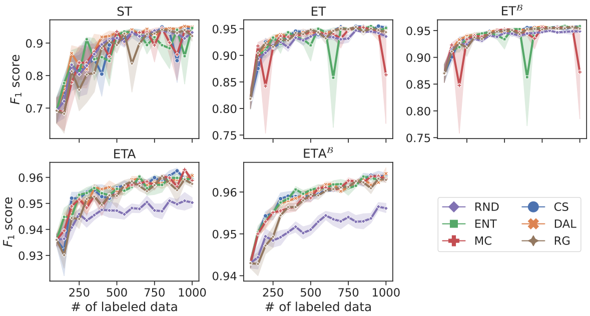
(a) TREC-2; BERT



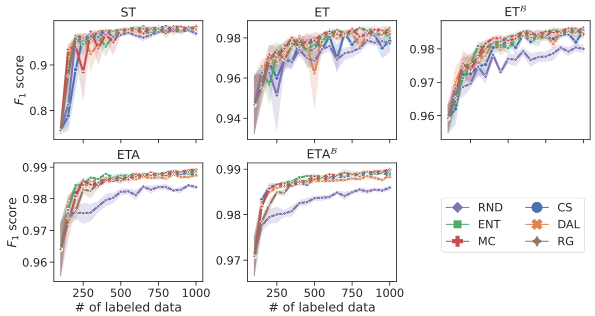
(b) TREC-2; ELECTRA



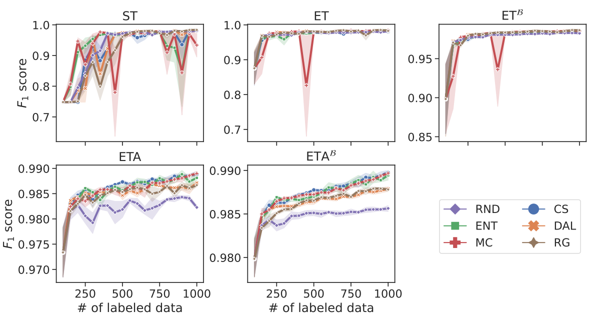
(c) SUBJ; BERT



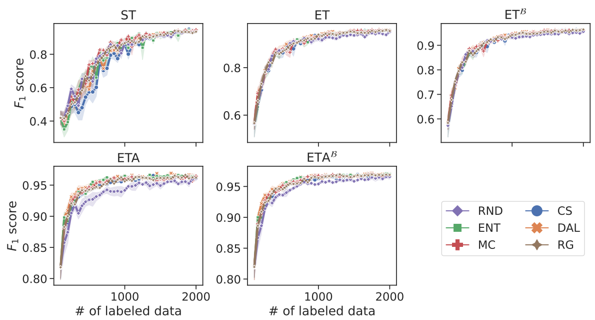
(d) SUBJ; ELECTRA



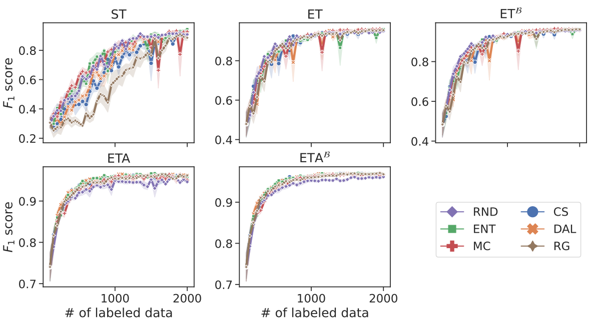
(e) AGN-2; BERT



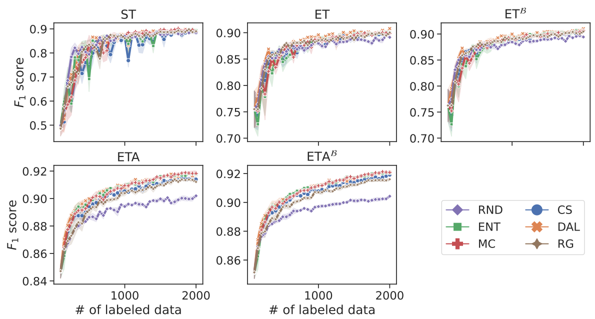
(f) AGN-2; ELECTRA



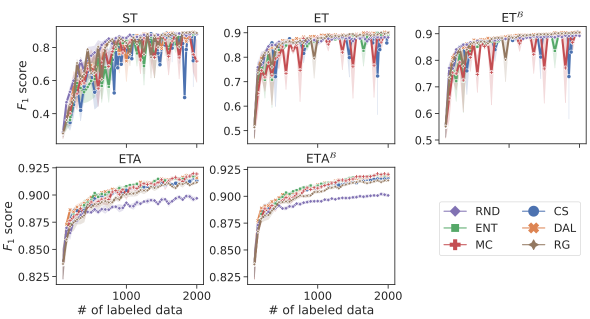
(g) TREC-6; BERT



(h) TREC-6; ELECTRA



(i) AGN-4; BERT



(j) AGN-4; ELECTRA

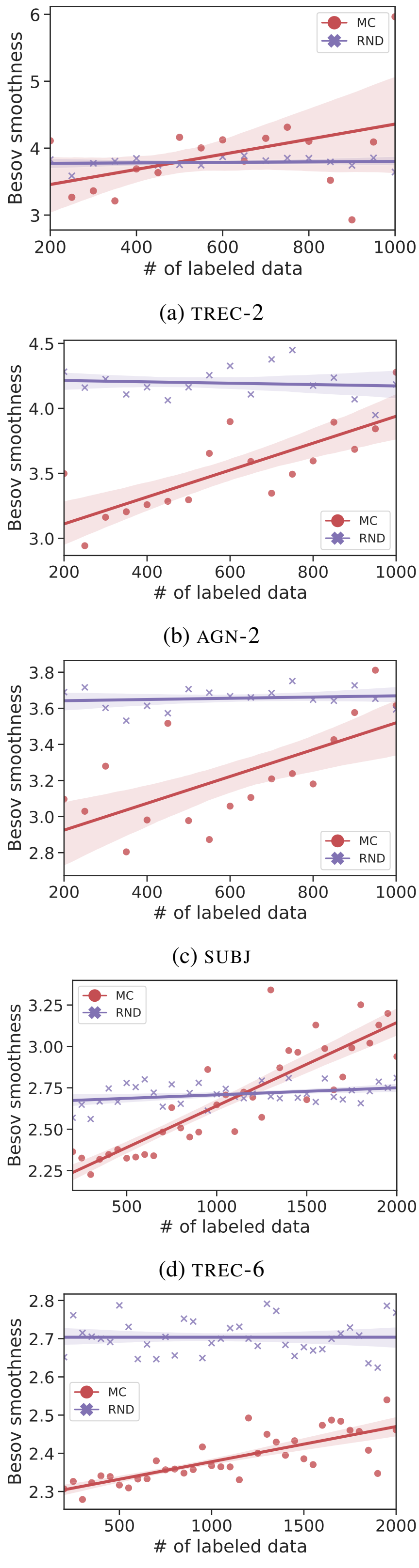
Figure 4: AL performance curves for different training regimes across datasets and models. Random sampling (purple rhombs) serves as a baseline. Best viewed on a computer screen.

		RND	ENT	MC	CS	DAL	RG
TREC-2	ST	.831	.829	.836	.835	.840	.805
	ET	.910	.924	.918	.928	.927	.919
	$\text{ET}^{\mathcal{B}}$	.919	.930	.926	.934	.936	.927
	ETA	.932	.953	.953	.953	.951	.949
	$\text{ETA}^{\mathcal{B}}$	.939	.959	.958	.959	.956	.956
SUBJ	ST	.880	.872	.870	.871	.898	.860
	ET	.926	.927	.925	.935	.936	.935
	$\text{ET}^{\mathcal{B}}$	.938	.937	.934	.944	.946	.942
	ETA	.946	.955	.954	.955	.955	.952
	$\text{ETA}^{\mathcal{B}}$	.952	.959	.959	.959	.959	.957
AGN-2	ST	.867	.901	.891	.850	.850	.823
	ET	.963	.963	.954	.963	.966	.965
	$\text{ET}^{\mathcal{B}}$	.969	.971	.962	.971	.971	.972
	ETA	.977	.981	.981	.982	.981	.980
	$\text{ETA}^{\mathcal{B}}$	.980	.983	.983	.983	.982	.982
TREC-6	ST	.604	.645	.636	.549	.561	.461
	ET	.837	.848	.839	.817	.811	.814
	$\text{ET}^{\mathcal{B}}$	.843	.858	.847	.821	.813	.816
	ETA	.897	.917	.905	.905	.905	.901
	$\text{ETA}^{\mathcal{B}}$	.906	.925	.917	.915	.914	.911
AGN-4	ST	.793	.706	.713	.688	.755	.750
	ET	.857	.844	.824	.845	.866	.857
	$\text{ET}^{\mathcal{B}}$	.868	.855	.836	.857	.874	.866
	ETA	.888	.903	.901	.901	.904	.897
	$\text{ETA}^{\mathcal{B}}$	.893	.907	.905	.905	.907	.901

Table 5: AUC for random sampling and different AL methods across datasets and training regimes for ELECTRA. The results are averaged over five runs with different seeds.

	ST	ET	ETA	$\text{ET}^{\mathcal{B}}$	$\text{ETA}^{\mathcal{B}}$
TREC-2	.0093	.0053	.0045	.0026	**.0022**
SUBJ	.0117	.0045	.0032	.0013	**.0008**
AGN-2	.0100	.0036	.0020	.0009	**.0005**
TREC-6	.0134	.0081	.0074	.0032	**.0027**
AGN-4	.0118	.0048	.0045	.0022	**.0014**

Table 6: Average standard deviation for different training regimes. The results are averaged across models and AL methods. **Bold** numbers indicate regimes with the lowest standard deviation for a particular dataset.



(a) TREC-2

(b) AGN-2

(c) SUBJ

(d) TREC-6

(e) AGN-4

Figure 5: Besov smoothness of actively acquired samples with MC (red) compared to the smoothness of random samples (purple).

Entrenchment Matters: Investigating Positional and Constructional Sensitivity in Small and Large Language Models

Bastian Bunzeck and **Sina Zarrieß**
Computational Linguistics
Bielefeld University, Germany
{bastian.bunzeck, sina.zarriess}@uni-bielefeld.de

Abstract

The success of large language models (LMs) has also prompted a push towards smaller models, but the differences in functionality and encodings between these two types of models are not yet well understood. In this paper, we employ a perturbed masking approach to investigate differences in token influence patterns on the sequence embeddings of larger and smaller RoBERTa models. Specifically, we explore how token properties like position, length or part of speech influence their sequence embeddings. We find that there is a general tendency for sequence-final tokens to exert a higher influence. Among part-of-speech tags, nouns, numerals and punctuation marks are the most influential, with smaller deviations for individual models. These findings also align with usage-based linguistic evidence on the effect of entrenchment. Finally, we show that the relationship between data size and model size influences the variability and brittleness of these effects, hinting towards a need for holistically balanced models.

1 Introduction

Recent years have witnessed an exponential growth in the size of language models, which has led to significant improvements in their performance on various natural language processing tasks. However, the reasons behind the remarkable success of LMs remain elusive, and it is questionable whether further growth will enhance their performance (Hong et al., 2022). More recently, it has been shown that small models can potentially learn linguistic structure equally well (Warstadt et al., 2020b; Zhang et al., 2021; Huebner et al., 2021). Because of neural network's opaque functionality, the reasons for these similarities and differences are not yet well understood. While grammatical evaluation suites (Warstadt et al., 2020a; Huebner et al., 2021; Newman et al., 2021) focus more on model output, evaluation approaches from the field of BERTology (Rogers et al., 2020) try to address this problem by studying the model's internal representations and mechanics. The present paper employs a perturbed masking approach (Wu et al., 2020) to study the influence of syntactic and constructional factors on lexical influence in sentence embeddings, and their differences between smaller and larger models.

To investigate these differences, we propose an approach inspired by usage-based linguistics. In the usage-based view, grammar is seen as emerging from language use and domain-general learning mechanisms (Diessel, 2019). Constructions, form-meaning pairings on all levels of linguistic analysis, are seen as the essential building blocks of language (Fillmore, 1988; Goldberg, 2003). Domain-general processes that influence such construction grammars are highly dependent on frequency effects in the input. For example, repeated use of a linguistic structure leads to it becoming more entrenched, more *unit-like*, in a speaker's "cognitive organization" (Langacker, 1987, 59). As artificial neural networks are domain-agnostic, statistical learners that create their linguistic systems through repeated use in the learning process, such effects attested in human language users should also be present in artificial learners. Consequently, usage-based approaches should be able to provide new insights on understanding the linguistic capabilities of language models, and the differences between large and small models (with different amounts of input) in particular.

Within our usage-based framework, we explore the influence that individual tokens have on the embeddings of their sequence. In opposition to grammatical test suites that challenge LLMs' abilities on very specific phenomena and structures, we aim to explore and analyze the linguistic abilities of LMs in terms of general positional and constructional factors and influences in their representations. Furthermore, by comparing these aspects for models trained on different amounts of

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 25–37
September 11–12, 2023.

linguistic data, we aim to find out whether, and if so, how fast constructional entrenchment and generalizations may arise. Because their representations are shaped by less input data, smaller models may exhibit less entrenchment effects, and be thus more brittle and sensitive. By comparing the most influential parts of speech for the models and construction types, we aim to find out on which grammatical categories the sequence embeddings depend the most and whether this is changed through the amount of training data. Finally, the frequency and information effects should affect the representations diametrically. The present analysis will tell if any effect is stronger in larger LMs.

Generally, we find that LMs of all sizes have a bias towards attributing more weight/influence to tokens at the end of a sentence, which aligns with the information-driven aspect of the linguistic theory. However, although there are construction-dependent differences, this effect does not vary systematically between the construction types. This is surprising, considering their differences in lexical specificity. Furthermore, we find that this bias is influenced by model size, but not in a linear fashion. Finally, we find that all models assign the highest importance to lexical words, especially nouns. In this sense, our results suggest that entrenchment as a property of statistical learners does indeed map from usage-based theories to artificial learners. Yet, its interplay with model structure and learning processes remains complex and not completely transparent.

2 Motivation for usage-based approaches to LMs

Established evaluation suites for grammatical abilities (Warstadt et al., 2020a; Huebner et al., 2021; Newman et al., 2021) often work by focusing on models' preferences for grammatical utterances over their ungrammatical counterparts. These techniques are inspired by a rather strict generative view of language, which assumes a pre-endowed human language faculty that generates grammatical strings of words from a hypothesized mental hierarchical structure (Chomsky, 1957, 1965). If the goal of neural language modelling was to recreate this, then only testing on phenomena like binding or filler-gap relations would be sufficient. However, alternative approaches to linguistic theory question these notions. The usage-based approach sees grammar as a fuzzy mental model of language that is constantly shaped and re-shaped by domain-general cognitive mechanisms, such as automatization, entrenchment or analogy, through input and usage (Tomasello, 2003; Diessel, 2019). The resulting mental representations in the form of linguistic constructions are influenced by frequency effects. For example, forms that are perceived and produced more often are more deeply entrenched in mental grammar (Schmid, 2015). On the syntactic level, such effects have syntagmatic and paradigmatic dimensions. The syntagmatic dimension refers to which elements occur sequentially, whereas the paradigmatic dimensions is concerned with the variation possible for certain positions/slots in syntagms. Constructions exhibit different levels of such variation. For example, wh-questions like *Where is the butter?* only have a limited number of options for the question word (first slot) or the auxiliary (second slot), whereas the last slot can be filled by any noun. These variation effects manifest in different phenomena. Research from child-directed speech shows that spoken language is organized around *lexical frames*, lexically restricted sentence beginnings that occur with a much higher frequency than their lexically diverse counterparts, and which differ for syntactic construction types (Cameron-Faulkner et al., 2003). While the possible linguistic variation in language production is quasi-infinite, speakers rely on highly frequent, mentally automatized combinations to initiate utterances. This preference is commonly related to ease of production, a factor that is also realized in phonetic reduction or syntactic contraction of high frequency units (Bybee and Thompson, 1997). Such producer-oriented factors grounded in automatization and entrenchment are, however, not the only usage-based variables shaping variation in utterance production. For example, the information weight principle (Behaghel, 1930; Quirk, 1972; Arnold et al., 2000) posits that new information and longer, "heavier" constituents in English are commonly placed at the end of utterances, which facilitates communicative ease from a hearer-oriented perspective. Such aspects only play a very minor role in current approaches to evaluating the grammatical abilities and behaviour of LMs, although they share many underlying concepts these models. Consequently, a new, usage-based paradigm to the evaluation and analysis of LLMs is needed, as it enables new insights that are not derivable from current approaches.

3 Related work

Although perturbed masking, the analysis of model architecture or size, and contructionist/usage-based approaches to NLP have had little to no overlap in previous research, they have been used for insightful analyses on their own. The following paragraphs offer a short review of the current literature in these research directions.

Perturbed masking Wu et al. (2020) show that perturbed masking can be used to retrieve dependency trees, constituency trees and document-level discourse structures by inducing tree structures from influence matrices, based on tokens connected by higher influence values. While not as exact as other parsing approaches, they showed that BERT-based representations already encode syntactic structure. Taktasheva et al. (2021) investigate the influence of syntactic perturbations through position shifts of syntactically grouped n-grams and clauses inside sentences for English, Swedish and Russian BERT models. They find that the perturbation patterns vary for languages with different degrees of word order flexibility and that syntactic representations can be better restored from languages with fixed word order (e.g. English). In an earlier approach, the NLIZE (Liu et al., 2019a) system visualized perturbation-based changes in attention heads and output weights for natural language inference tasks. The approach has not yet been used for construction-oriented analyses or the investigation of smaller models.

Model size Differences between smaller and larger models have only begun to get systematically investigated, and existing studies have arrived at somewhat contradictory conclusions. Warstadt et al. (2020b) trained a variety of RoBERTa models with growing amounts of data, ranging from 1M to 1B tokens. They show that only larger models begin to exhibit preferences for linguistic generalizations over surface generalizations. The additional amount of data appears crucial for this difference. In contrast, the BabyBERTa model (Huebner et al., 2021) restricts the model size, number of intermediate layers and attention heads. Its training data is comparably small and was sampled from child-directed speech from the CHILDES corpora. Despite these limitations, its performance across their own evaluation suite, Zorro, is similar to the much larger RoBERTa-base model, questioning if ever larger amounts of data are actually needed, or whether the combination of hyperparameters and training data is actually responsible for emergent generalizations.

Construction grammar More recently, LLMs have also begun to be investigated from a construction grammar viewpoint. Tayyar Madabushi et al. (2020) show through a series of probing experiments that BERT embeddings already contain information that could be seen as constructionist, for example by being able to successfully determine whether two sentences with little to no lexical overlap instantiate the same grammatical construction. Tseng et al. (2022) fine-tune a BERT model for a cloze completion task on open slots in Taiwanese Mandarin constructions and show that it improves performance. Moreover, sentences that instantiate the same construction tend to be spatially closer in the vector space than sentences with different constructions but the same main verb (Li et al., 2022). However, it remains questionable how applicable such knowledge is, as Weissweiler et al. (2022) find that LLMs fail to deduce conclusions from the comparative correlative construction in an inference task. Finally, Weissweiler et al. (2023) summarize the previous line of constructionist inquiry into LLMs. They find that current research has focused on only a very limited set of constructions and that there are differences in what is assumed to be evidence for the presence of constructionist information in LLMs. Consequently, they call for a diversification of constructionist research in terms of data sources and methodology. The present paper responds to this call by investigating constructions as processing units and their influence on sequence embeddings. By employing constructions as an additional analytical factor, not as the end point of the analysis, we expand on this previous research.

4 Methods

4.1 Perturbed masking

We use Wu et al.'s (2020) perturbed masking approach to calculate the influence of a token x on its sequence. This approach is adequate as a measure of influence because it captures the global influence patterns between all token pairs in a sequence, and not only the influence of one token on the entirety of a sequence.

1. For each other token y in the sequence:
   (a) y is replaced with the `<mask>` token

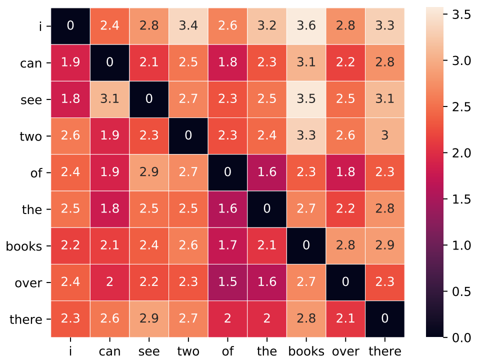


Figure 1: Influence heatmap for the sentence *I can see two of the books over there* encoded with roberta-base

(b) the sequence embedding s_y with masked y is computed

(c) x is additionally replaced with the `<mask>` token

(d) the embedding $s_{x,y}$ for the sequence with both x and y masked is calculated

(e) the vector distance d between s_y and $s_{x,y}$ is calculated to measure the influence of x on the embedding of y

2. By averaging the distances d between token x and all other tokens y, an average influence value of x on the sequence is acquired

We use the penultimate layer of the BERT models as the source of our embeddings, as Devlin et al. (2019) report that these embeddings perform consistently well across a variety of tasks.

By examining how certain tokens impact their embeddings more significantly, we can assess the degree to which these tokens become deeply embedded (or *entrenched*) in the learning model. Tokens with a stronger influence on their embeddings are indicative of greater entrenchment, reflecting their increased importance and resistance to modification within LMs.

4.2 Test data

We chose naturally occurring sentences from the CHILDES family of corpora (MacWhinney, 2000) as the basis of our analysis. They are especially suited to this experimental setup, because the vocabulary of child-directed speech is restricted to fairly frequent words that should be present in all models' training data, and the individual sentences are rather short and syntactically not overly complex, yet grammatical. Due to the uniform nature of child-directed speech, we also control for the influence of highly unusual or infrequent words that could disproportionately affect the perturbation data.

We sampled a data set of 3.000 test sentences from the English section of CHILDES – 1.000 sentences per construction type of interest. They were retrieved from the corpus through pattern matching on part-of-speech-tagged data. We annotated the CHILDES data with 14 different construction types inspired by Cameron-Faulkner et al. (2003) with a construction parser that operates on word class patterns. We chose three focus constructions[1] – imperatives, wh-questions and transitive sentences. We decided on these construction types because they differ in their word order (and its strictness), as well as in their lexical variation (Cameron-Faulkner et al., 2003). While transitives, for example, have a near-infinite amount of possible beginnings, wh-questions are constrained to the word class of interrogatives. This variation in syntactic and lexical patterns should shed additional light on positional and other entrenchment effects – focusing on one construction type only could taint the results by being biased from these factors.

Our parser retrieved the construction types with an accuracy of over 93% compared against a manually annotated ground-truth data set. For each construction type of interest, we then sampled 1.000 sentences randomly. To reduce variation introduced by different sentence lengths or patterns of clausal combination, only utterances with nine tokens or less were considered. The mean utterance length lies a little below that (M = 5.84, SD = 1.38), with wh-questions being the shortest (M = 5.20, SD = 1.54), transitive sentences being the longest (M = 6.28, SD = 1.25) and imperatives in between (M = 6.06, SD = 1.06).

4.3 Models

To maintain comparability of model architecture, we exclusively analysed models with RoBERTa architectures. These include the two original roberta-base[2] and roberta-large (Liu et al., 2019b) models, the distilled distilroberta-base (Sanh et al., 2020) as well as models trained with different amounts of input by Warstadt et al. (2020b). The model properties are compared in Table 1. The training data for all models was sourced from a combina-

[1]The respective patterns for the three constructions can be found in appendix A.

[2]For the rest of this paper, we denote the models by their lowercase names.

	Hidden layers	Parameters	Attention heads	Embedding size	Training data
roberta-base	12	125M	12	768	30B tokens
roberta-large	24	355M	16	1024	30B tokens
distilroberta-base	6	82M	12	768	30B tokens
roberta-med-small-1M	6	45M	8	512	1M tokens
roberta-base-10M	12	125M	12	768	10M tokens
roberta-base-100M	12	125M	12	768	100M tokens
roberta-base-1B	12	125M	12	768	1B tokens

Table 1: Hyperparameters of target models

tion of BookCorpus, English Wikipedia, CC-News, OpenWebText, and Stories dataset, and sampled down for the smaller models.

Although all of these eight models are large LMs, some of them can be considered "small" for the sake of the present analysis. roberta-med-small-1M features a reduced architecture and smaller dataset (1M tokens), whereas roberta-med-small-10M only has a smaller dataset (10M tokens) but no reduced architecture. Finally, distilroberta-base has a smaller architecture that was later fine-tuned to mirror the larger roberta-base (based on 30B tokens). Consequently, the first two models can be considered small in terms of architecture and data, whereas the latter are small in data or architecture. BabyBERTa was excluded from the analysis because its training data contains a part of the test data we used.

4.4 Experimental setup

For all 3.000 sentences and all 7 models, we perform perturbed masking with the transformers library (Wolf et al., 2020). Figure 1 shows the result of one perturbation run on the sequence *I can see two of the books over there*. The influencing words are shown on the x-axis, the influenced words on the y-axis. A higher numerical value, shown by brighter-colored cells, stands for a larger vector distance between the once and twice masked sequence embeddings. This indicates that the respective influencing word exerts a higher influence on the embedding of the influenced word – it changes its numerical values more strongly. As Wu et al. (2020) note, these patterns often align with grammatical relations. For example, *books* as the grammatical object here exerts the most influence on *I*, *see* and *two* – the subject and predicate in the sentence and a numeral that defines it.

We average the influence values per column, which gives a measure of the average influence a token exerts on its sentence embedding. For each token, we store this influence value together with its part-of-speech tag, tagged with spaCy (Honnibal et al., 2020). For all LMs, we fit a linear regression model for the influence value as a dependent variable, with the following independent variables[3] with statsmodels (Seabold and Perktold, 2010):

- Token position, to investigate whether a position bias exists
- Token length (in characters), to see how different parts of speech and/or higher information content affect the influence
- Sequence length (number of tokens), to see how longer sequences affect influence values
- Construction type, to see whether paradigmatic and syntagmatic differences mediate these effects

Furthermore, we calculate the average influence for part-of-speech categories for all model/construction combinations.

5 Results

5.1 Regression analysis

Table 2 shows the linear models for each investigated language model, reporting the intercept and the regression coefficients for token position, word length, sentence length, the construction type, and the R^2 for the respective regression. For the construction type, imperative sentences form the baseline, while the other two types were included as categorical variables, which means that their corresponding results signify their relative impact compared to the influence values for the imperative sentences. The test statistics of all models' F-tests, as well as those of the t-tests for all values, were statistically highly significant ($p < 0.001$). We acknowledge that this significance might also be caused by the high number of data points available.

[3]Position, token length and sequence length were normalized to values in the range $[0; 1]$ before fitting the linear regression models.

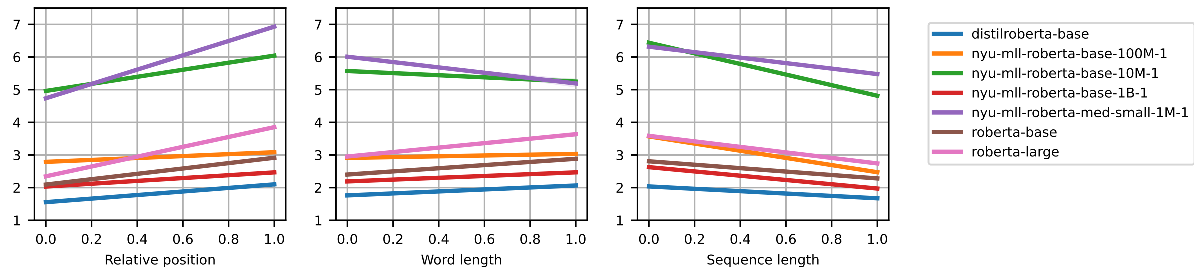


Figure 2: Regression lines for token position, word length and sentence length (calculated independently), with the influence value as the dependent variable

	roberta-med-small-1M-1	roberta-base-10M-1	roberta-base-100M-1	roberta-base-1B-1	roberta-base	roberta-large	distilroberta-base
(intercept)	4.871	5.367	3.188	2.115	2.012	2.186	1.478
token position	2.351	1.225	0.373	0.557	1.009	1.818	0.675
word length	1.075	0.978	0.538	0.834	1.287	2.129	0.883
sentence length	-0.932	-1.959	-1.210	-0.712	-0.550	-0.909	-0.402
cxn = wh-question	0.158	0.847	0.263	0.216	0.057	0.123	0.122
cxn = transitive	0.206	0.582	0.208	0.109	0.090	0.190	0.090
R^2	0.372	0.506	0.388	0.264	0.373	0.409	0.428

Table 2: Summary of parameter estimates for the logistic regression models across RoBERTa models of different size (column order roughly corresponds to model size). The response variable corresponds to the calculated influence values per word. The baseline for the categorical variable “construction” (*cxn*) are the imperative sentences included in the data set.

Across all models and constructions, the **token position** has a positive effect on the influence value. The same can be said for **word length**. This means that, for the current experiment, small and large LMs have a clear and systematic preference for putting more weight on sequence endings and longer words. This effect, to the best of our knowledge, has not been described before. Interestingly, for the two smallest models, the regression coefficient is larger for token position than for word length, a relation that is reversed for all other models trained on more data. **Sentence length** has a constantly negative effect on the influence value – longer sequences thus lower the influence values of their contained tokens. The effects of **construction type** on the influence value are generally positive, which points towards individual words' influence being higher in transitive sentences and wh-questions, when compared to the baseline (imperatives). These tendencies with regard to constructions are stable across all models.

Figure 2 shows model-wise regression lines only incorporating token position *or* word length *or* sentence length. The two models trained with the least amount of data (1M and 10M tokens) have the highest absolute influence values. This is also reflected in the comparatively high intercepts reported in Table 2. The other models' values are roughly equivalent. Interestingly, word length alone has a slightly negative effect for the two smallest models (1M and 10M). When incorporating all variables in the regression model, as in Table 2, this tendency is reversed. It is plausible that other interaction effects exist between the independent variables, which further underlines the importance of accounting for all of them in the regression model.

Finally, the R^2-values as goodness-of-fit measures exhibit considerable variation between the models. Training data and model size appear to play a certain role, as roberta-base-10M, the model with the least amount of data learned with a non-reduced architecture, features the highest R^2-value. However, the 10M model also features (by far) the largest regression coefficients for the construction types. Overall, the values range from 26.4% for the roberta-base-1B model to 50.6% for the 10M model. This shows that, although position, word length, sentence length and construction type cannot function as the sole predictors of influence values, they are influential variables under certain circumstances, possibly mediated by architectural factors beyond the scope of the present analysis.

Crucially, there is no linear relationship between the amount of training data, the model size and the goodness-of-fit, hinting towards an interconnectedness and/or more or less “fitting” combinations of internal factors like number of layers or attention heads, and model size.

5.2 Impact of construction type on part-of-speech influence

General overview In addition to the regression analysis, we calculated the average influence values on the part-of-speech level. These values are shown in Tables 3, 4 and 5, with the top three/four highest values set apart in bold face. Across all construction-model combinations, nouns, proper nouns and punctuation symbols are consistently influential.[4] For the models trained on more data, verbs are quite influential as well (at least for transitive sentences and wh-questions). For wh-questions and imperatives, also numerals are sporadically in the top three. Overall, the three different construction types show a similar, but still variable picture. Regarding model size, it is interesting to note that the smallest and the largest models (and distilroberta-base) tend to have their own, fairly stable rankings, whereas the most outliers occur in the medium-sized models.

Wh-questions The data for the wh-question in Table 3 is the least straightforwardly interpretable – only proper nouns are consistently influential. Apart from that, a contrast between the smallest/largest models and the medium-sized ones is noticeable. The smallest/largest models feature nouns and punctuation marks as most influential, whereas the medium models show larger influence values for numerals, verbs, and also (once) for adverbs and auxiliaries.

Transitive sentences For the transitive sentences (Table 4), nouns and punctuation marks are constantly among the most influential parts of speech. Here, a division can be drawn between the smaller and larger models. Whereas smaller models focus more on proper nouns, the larger models feature high influence values for verbs. For the roberta-base-1B model, one outlier is the auxiliary tag already found in the wh-question data.

[4]The X tag for unknown part of speech was consistently strong as well. A closer inspection of the dataset yields that the respective tokens are family-internal onomatopoeia or similar phonetic descriptions, which most probably are absent from the training data and thus *qua definitionem* more influential. Consequently, they were not set apart in bold face.

Imperatives The data for imperative sentences in Table 5 once more features punctuation and nouns as the most influential. Proper nouns are also highly influential, except for the two medium-sized (100M and 1B) models, where pronouns and auxiliaries also play a role.

6 Discussion

The present analysis in section 5.1 has shown that token position, word length and sequence length strongly affect sequence embeddings in terms of the influence of their lexical elements (viz. tokens). The general effect of token position on token influence is positive and stable across seven different RoBERTa models. All models exhibit a sequence-ending bias for the influence values. However, the effects exhibit variable strength. One reason for this could lie in the model size (training data, hyperparameters and model internals) – the effects (and the absolute influence values) are higher for smaller models but also for the largest model.

Apart from the token position, word length also has a constantly positive effect on the influence value. Longer words are thus more influential. The Zipfian law (Zipf, 1935) posits an inverse relationship between word frequency and length. As the most frequent words tend to be function words, the positive effect of word length on the influence value could also hint towards the higher informational content of longer words. Piantadosi et al. (2011) find a high correlation between word length and informational content for words in English, Swedish and German.

The negative impact of the sentence length could be caused by more tokens having to “share” the informational content of the whole sequence, which is then divided between all of them. The positive influence of the non-imperative sentences allows a ranking of construction type influence, where the effect is the weakest for imperatives, stronger for transitives and the strongest for wh-questions. From a linguistic point-of-view, the reasons for this remain elusive. Wh-questions and imperatives have more syntactically and lexically-fixed constructional schemas than transitive sentences. There is a possible connection between the functional aspects of imperatives and their reduced influence values, because they usually trigger real-world actions. In contrast, the information-driven functions of (information-demanding) questions and (information-conveying) regular transitive sentences could impact their tokens' influence values positively.

When directly comparing the parts of speech, construction types do not seem to hold much explanatory value either. There is no systematic variation between their preferred parts of speech. In contrast, nouns, proper nouns, numerals and punctuation marks are consistently important across the types. Outliers are sporadic, only the higher influence of verbs in transitive sentences embedded through models with more training data is somewhat systematic. Importantly, including positional information is an active research topic in contemporary NLP (see Dufter et al., 2022 for a survey). The present results suggest that the token influence patterns already encode positional information, although transformers are theoretically invariant to the reordering of tokens in a sequence.

Comparing the data from a model-oriented perspective yields interesting, although ambiguous and inconclusive results. The smallest models (in terms of architecture and training data), as well as the largest models (in terms of training data) stabilize in different ways with regard to their most influential parts of speech. The medium-sized models (in terms of training data) exhibit more variation, focus on more exotic parts of speech and the corresponding linear regressions have a somewhat lower goodness-of-fit as well as lower overall regression coefficients. Crucially, it seems that for a stable and predictable functionality, a certain match between model size in terms of internal architecture (hidden layers, attention heads, etc.) is needed. Small data needs smaller models, and large data needs larger models. If these factors do not match, the representations become brittle and potentially less useful for downstream tasks. The concrete make-up of such matching combinations still needs more empirical scrutiny. For example, the model with the highest R^2, roberta-base-10M, also features the highest regression coefficients for the construction types. This relationship does not stabilize across the other model-data combinations, with no discernible reasons identifiable from the present analysis.

Also, as further empirical results show that the processing in LMs mirrors traditional NLP pipelines along the layers of linguistic processing (Tenney et al., 2019), the value of LMs for studies of linguistic processing has been put to question (Linzen and Baroni, 2021; Warstadt and Bowman, 2022). Pannitto and Herbelot (2022) argue that neural networks should also be used to investigate usage-based theories of language. The present study has added to this emergent field by showing that findings from usage-based linguistics on the importance of sequence order to language use are indeed mirrored in transformer-based LMs. However, the construction-level effects proposed in linguistic literature could not be completely verified. This might be due to the very different nature of language acquisition in humans and the training procedure in ANNs. Training only mirrors the frequency-driven aspect of usage-based linguistics. Other aspects like embodied cognition or the functional dimension of language, which can also be linked to construction types (e.g. in Cameron-Faulkner and Hickey (2011)), are missing. Remarkably, function words are not as influential as lexical words. Their structural predictability could be an influence factor in this case. Constructions are usually conceptualized as structures with open slots. Here, paradigmatic variation is much higher for lexical words, which are also more influential for models. However, the great amount of variation suggests that not all LMs learn the exact same structures, with inadequate data/model matchings leading to more brittle representations. Dąbrowska (2012) argues that the grammatical systems of adult speakers do not completely align with each other – they are only similar enough to enable effective communication. Judging from our results, the grammatical systems in language also feature different sensitivities to factors like word length or sequence length. This could point to learning with fitting parameter combinations being more human-adequate, as the linguistic and architectural effects on LMs are gradient in nature (a feature they share with human language processing and usage). Most importantly, this analysis has shown that the trade-offs between data size, model internals, and stable performance deserve further recognition and investigation, because mismatched combinations may lead to unstable or brittle representations.

7 Conclusion

Our investigation shed light on the functionality of LMs from a usage-based perspective, and has shown that concepts from usage-based linguistics, like entrenchment, can be used fruitfully in the analysis of such LMs. We discovered that frequency-driven factors, as well as information weight, play a significant role in these models' encodings. Notably, the models exhibit a bias towards the ends of sequences, with the influence of tokens positively correlated with their length and information-rich parts of speech, such as nouns. However, these effects weaken in longer sequences. The high variation across our statistical models' R^2-values hints at additional factors beyond entrenchment being at play when determining token influence on sequence embeddings. Still, our findings suggest that human learners and artificial learners share similarities, as both processes are influenced by frequency and information effects. Significant differences in influence values between construction types indicate a need for further research to interpret these differences linguistically. Additionally, our study explored the similarities and differences between models trained with varying amounts of data. While general effects remain similar, there is increased volatility, especially in preferred parts of speech, with shrinking data size. A non-linear relationship between the amount of training data, model architecture, and effect sizes/goodness-of-fit was observed. This highlights the need for deeper investigations into the optimal combinations of data and other hyperparameters.

Limitations

The present study is limited by the availability of models with different, yet comparable (e.g. in terms of training data or traceable stepwise adjustment) training regimens. More empirical results with regard to data size and model internals, investigated in a systematic and controlled way, are clearly needed. Furthermore, it would also be interesting to additionally look into the perturbation patterns for different layers in LMs, which could further illuminate the ways in which structural sensitivity mirrors the levels of human linguistic processing.

Acknowledgements

We acknowledge financial support by the project "SAIL: SustAInable Life-cycle of Intelligent Socio-Technical Systems" (Grant ID NW21-059A), which is funded by the program "Netzwerke 2021" of the Ministry of Culture and Science of the State of Northrhine Westphalia, Germany.

References

Jennifer E. Arnold, Anthony Losongco, Thomas Wasow, and Ryan Ginstrom. 2000. Heaviness vs. newness: The effects of structural complexity and discourse status on constituent ordering. *Language*, 76(1):28–55.

Otto Behaghel. 1930. Von deutscher Wortstellung. *Zeitschrift für Deutschkunde*, 44:81–89.

Joan Bybee and Sandra Thompson. 1997. Three Frequency Effects in Syntax. *Annual Meeting of the Berkeley Linguistics Society*, 23(1):378.

Thea Cameron-Faulkner and Tina Hickey. 2011. Form and function in Irish child directed speech. *Cognitive Linguistics*, 22(3):569–594.

Thea Cameron-Faulkner, Elena Lieven, and Michael Tomasello. 2003. A construction based analysis of child directed speech. *Cognitive Science*, 27(6):843–873.

Noam Chomsky. 1957. *Syntactic Structures*. Mouton Publishers.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. Number 11 in Massachusetts Institute of Technology. Research Laboratory of Electronics. Special Technical Report. MIT Press, Cambridge, Massachusetts.

Ewa Dąbrowska. 2012. Different speakers, different grammars: Individual differences in native language attainment. *Linguistic Approaches to Bilingualism*, 2(3):219–253.

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*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Holger Diessel. 2019. Usage-based construction grammar. In Ewa Dąbrowska and Dagmar Divjak, editors, *Cognitive Linguistics - A Survey of Linguistic Subfields*, pages 50–80. De Gruyter.

Philipp Dufter, Martin Schmitt, and Hinrich Schütze. 2022. Position Information in Transformers: An Overview. *Computational Linguistics*, 48(3):733–763.

Charles J. Fillmore. 1988. The Mechanisms of "Construction Grammar". *Annual Meeting of the Berkeley Linguistics Society*, 14:35.

Adele E. Goldberg. 2003. Constructions: A new theoretical approach to language. *Trends in Cognitive Sciences*, 7(5):219–224.

Zhi Hong, Aswathy Ajith, Gregory Pauloski, Eamon Duede, Carl Malamud, Roger Magoulas, Kyle Chard, and Ian Foster. 2022. ScholarBERT: Bigger is Not Always Better.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in python.

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. BabyBERTa: Learning More Grammar With Small-Scale Child-Directed Language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. Association for Computational Linguistics.

Ronald W. Langacker. 1987. *Foundations of Cognitive Grammar. Vol. 1: Theoretical Prerequisites*, volume 1. Stanford University Press, Stanford, California.

Bai Li, Zining Zhu, Guillaume Thomas, Frank Rudzicz, and Yang Xu. 2022. Neural reality of argument structure constructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7410–7423, Dublin, Ireland. Association for Computational Linguistics.

Tal Linzen and Marco Baroni. 2021. Syntactic Structure from Deep Learning. *Annual Review of Linguistics*, 7(1):195–212.

Shusen Liu, Zhimin Li, Tao Li, Vivek Srikumar, Valerio Pascucci, and Peer-Timo Bremer. 2019a. NLIZE: A Perturbation-Driven Visual Interrogation Tool for Analyzing and Interpreting Natural Language Inference Models. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):651–660.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Brian MacWhinney. 2000. *The CHILDES Project: Tools for Analyzing Talk*, 3 edition. Lawrence Erlbaum Associates, Mahwah, NJ.

Brian MacWhinney. 2018. MOR Manual.

Benjamin Newman, Kai-Siang Ang, Julia Gong, and John Hewitt. 2021. Refining Targeted Syntactic Evaluation of Language Models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3710–3723, Online. Association for Computational Linguistics.

Ludovica Pannitto and Aurelie Herbelot. 2022. Can Recurrent Neural Networks Validate Usage-Based Theories of Grammar Acquisition? *Frontiers in Psychology*, 13:741321.

Steven T. Piantadosi, Harry Tily, and Edward Gibson. 2011. Word lengths are optimized for efficient communication. *Proceedings of the National Academy of Sciences*, 108(9):3526–3529.

Randolph Quirk, editor. 1972. *A Grammar of Contemporary English*. Longman, London.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter.

Hans-Jörg Schmid. 2015. A blueprint of the Entrenchment-and-Conventionalization Model. *Yearbook of the German Cognitive Linguistics Association*, 3(1).

Skipper Seabold and Josef Perktold. 2010. Statsmodels: Econometric and Statistical Modeling with Python. In *Python in Science Conference*, pages 92–96, Austin, Texas.

Ekaterina Taktasheva, Vladislav Mikhailov, and Ekaterina Artemova. 2021. Shaking Syntactic Trees on the Sesame Street: Multilingual Probing with Controllable Perturbations. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 191–210, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Harish Tayyar Madabushi, Laurence Romain, Dagmar Divjak, and Petar Milin. 2020. CxGBERT: BERT meets Construction Grammar. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4020–4032, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT Rediscovers the Classical NLP Pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Michael Tomasello. 2003. *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press.

Yu-Hsiang Tseng, Cing-Fang Shih, Pin-Er Chen, Hsin-Yu Chou, Mao-Chang Ku, and Shu-Kai Hsieh. 2022. CxLM: A construction and context-aware language model. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6361–6369, Marseille, France. European Language Resources Association.

Alex Warstadt and Samuel R. Bowman. 2022. What Artificial Neural Networks Can Tell Us About Human Language Acquisition.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020a. BLiMP: The Benchmark of Linguistic Minimal Pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020b. Learning Which Features Matter: RoBERTa Acquires a Preference for Linguistic Generalizations (Eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online. Association for Computational Linguistics.

Leonie Weissweiler, Taiqi He, Naoki Otani, David R. Mortensen, Lori Levin, and Hinrich Schütze. 2023. Construction Grammar Provides Unique Insight into Neural Language Models. In *Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023)*, pages 85–95.

Leonie Weissweiler, Valentin Hofmann, Abdullatif Köksal, and Hinrich Schütze. 2022. The better your syntax, the better your semantics? Probing pretrained language models for the English comparative correlative. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10859–10882, Abu Dhabi, United Arab Emirates. 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.

Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. 2020. Perturbed Masking: Parameter-free Probing for Analyzing and Interpreting BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4166–4176, Online. Association for Computational Linguistics.

Yian Zhang, Alex Warstadt, Xiaocheng Li, and Samuel R. Bowman. 2021. When Do You Need Billions of Words of Pretraining Data? 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 1112–1125, Online. Association for Computational Linguistics.

George K. Zipf. 1935. *The Psycho-Biology of Language*. Houghton Mifflin, Boston.

A Construction retrieval patterns

The corpus files in CHILDES come with morphological annotation in form of the CHILDES tag set (MacWhinney, 2018, 8). To match the constructions, the following (high-level) patterns were considered and implemented via regular expressions:

- **Imperatives:** [adverb] + verb/modal + [quantifier] + [adverb] + [determiner] + noun/pronoun
- **Wh-questions:** [preposition] + interrogative pronoun/interrogative determiner/relative pronoun/conjunction + ?
- **Transitive sentences:** personal pronoun/subject pronoun/indefinite pronoun/noun/proper noun/demonstrative determiner + [modal/auxiliary] + [adverb] + [quantifier] + personal pronoun/subject pronoun/indefinite pronoun/noun/proper noun/demonstrative determiner

NB: The quality and consistency of morphological annotations varies considerably between the CHILDES data sets. Consequently, some part-of-speech tags had to be included for the pattern matching that make the constructional schemas appear intuitively wrong. Furthermore, questions were pre-annotated with question marks.

B Full influence values for all construction types

part of speech	roberta-med-small-1M	roberta-base-10M	roberta-base-100M	roberta-base-1B	roberta-base	roberta-large	distilroberta-base
ADJ	5.4	5.78	3	2.47	2.69	3.62	1.96
ADP	5.28	5.09	2.9	2.32	2.42	3.12	1.85
ADV	5.66	5.64	**3.24**	2.47	2.68	3.54	1.97
AUX	5.45	5.62	3.06	**2.81**	2.52	2.88	1.8
CCONJ	4.47	4.25	2.07	1.05	1.26	1.39	0.94
DET	4.56	4.97	2.76	2.1	2.06	2.45	1.58
INTJ	4.13	4.35	2.42	1.22	1.45	1.44	1.19
NOUN	**5.71**	5.77	3	2.4	**2.76**	**3.66**	**2.03**
NUM	5.7	**5.88**	**3.17**	**2.56**	2.66	**3.74**	1.95
PART	5.47	4.65	2.63	2.09	2.21	2.87	1.65
PRON	4.64	5.44	2.82	1.98	2.08	2.26	1.6
PROPN	**6.37**	**6.57**	**3.46**	**2.56**	**2.76**	**3.73**	**2.14**
PUNCT	**8.27**	**6.74**	3.1	2.46	2.73	3.56	**2.07**
SCONJ	4.12	4.89	2.63	1.32	1.76	1.41	1.42
VERB	5.54	5.85	3.11	**2.6**	**2.77**	3.6	1.98
X	7.15	6.96	3.36	3.09	3.16	3.92	2.29
(mean)	5.50	5.53	2.92	2.22	2.37	2.95	1.78

Table 3: Average influence values for wh-questions

part of speech	roberta-med-small-1M	roberta-base-10M	roberta-base-100M	roberta-base-1B	roberta-base	roberta-large	distilroberta-base
ADJ	5.93	5.36	2.79	2.26	2.57	3.35	1.9
ADP	5.59	4.64	2.65	2.04	2.32	2.99	1.79
ADV	5.85	5.25	2.78	2.25	2.47	3.15	1.83
AUX	5.71	5.24	2.77	**2.31**	2.28	2.89	1.7
CCONJ	4.62	4.48	2.4	1.23	1.39	1.55	1.09
DET	5.34	4.68	2.63	2.02	2.12	2.64	1.6
INTJ	4.55	4.49	2.4	1.36	1.53	1.57	1.23
NOUN	**6.38**	**5.69**	**2.91**	**2.3**	**2.8**	**3.54**	**2.03**
NUM	5.86	5.3	2.69	2.26	2.52	3.18	1.81
PART	5.33	4.18	2.39	1.85	2.09	2.49	1.65
PRON	5.32	5.07	2.68	1.96	2.24	2.49	1.56
PROPN	**6.6**	**6.02**	**3.09**	2.23	2.61	3.2	1.91
PUNCT	**6.44**	**6.13**	**3.09**	2.16	**2.66**	**3.41**	**1.92**
SCONJ	4.52	4.36	2.44	1.47	1.67	1.63	1.29
VERB	5.59	5.36	2.92	**2.46**	**2.65**	**3.42**	**1.92**
X	7.19	6.1	2.98	2.41	2.88	3.6	2.11
(mean)	5.68	5.15	2.73	2.04	2.30	2.82	1.71

Table 4: Average influence values for transitive sentences

part of speech	roberta-med-small-1M	roberta-base-10M	roberta-base-100M	roberta-base-1B	roberta-base	roberta-large	distilroberta-base
ADJ	5.87	5.25	2.93	2.32	2.75	3.44	1.93
ADP	5.74	4.86	3.03	2.36	2.55	3.36	1.87
ADV	6.03	5.4	**3.05**	2.35	2.64	3.36	1.88
AUX	5.72	5.26	2.69	2.08	2.24	2.81	1.68
CCONJ	5.16	4.74	2.65	1.83	1.92	2.44	1.35
DET	5.52	4.8	2.89	2.3	2.33	2.85	1.71
INTJ	4.99	4.76	2.5	1.55	1.73	1.89	1.37
NOUN	**6.73**	**5.71**	**3.06**	**2.43**	**3.09**	**3.89**	**2.16**
NUM	5.96	5.45	2.92	**2.47**	2.73	**3.57**	1.9
PART	5.44	4.33	2.61	1.94	2.12	2.48	1.54
PRON	5.67	5.23	2.91	**2.43**	2.56	3.22	1.78
PROPN	**6.89**	**5.96**	2.94	2.34	**2.9**	**3.66**	**2.05**
PUNCT	**6.75**	**6.19**	**3.22**	2.29	**2.76**	**3.57**	**1.98**
SCONJ	5.26	4.77	2.75	2.17	2.48	3.07	1.75
VERB	4.91	5.15	2.95	2.17	2.29	2.57	1.67
X	7.6	6.26	3.45	2.79	3.15	4.03	2.36
(mean)	5.89	5.26	2.91	2.24	2.52	3.14	1.81

Table 5: Average influence values for imperatives

Facilitating learning outcome assessment– development of new datasets and analysis of pre-trained language models

Akriti Jindal
Lakehead University
Thunder Bay, Canada
ajindal@lakeheadu.ca

Kaylin Kainulainen
Lakehead University
Thunder Bay, Canada
kkainula@lakeheadu.ca

Andrew Fisher
Saint Mary's University
Nova Scotia, Canada
andrew.fisher@smu.ca

Vijay Mago
York University
Toronto, Canada
vmago@yorku.ca

Abstract

Student mobility reflects academic transfer from one postsecondary institution to another and facilitates students' educational goals of obtaining multiple credentials and/or advanced training in their field. This process often relies on transfer credit assessment, based on the similarity between learning outcomes, to determine what knowledge and skills were obtained at the sending institution as well as what knowledge and skills need to still be acquired at the receiving institution. As human evaluation can be both a challenging and time-consuming process, algorithms based on natural language processing can be a reliable tool for assessing transfer credit. In this article, we propose two novel datasets in the fields of Anatomy and Computer Science. Our aim is to probe the similarity between learning outcomes utilising pre-trained embedding models and compare their performance to human-annotated results. We found that ALBERT, MPNeT and DistilRoBERTa demonstrated the best ability to predict the similarity between pairs of learning outcomes. However, Davinci - a GPT-3 model which is expected to predict better results - is only able to provide a good qualitative explanation and not an accurate similarity score. The codes and datasets are available at https://github.com/JAkriti/New-Dataset-and-Performance-of-Embedding-Models.

1 Introduction

Student mobility refers to the movement - or, "transfer" – of students from one post-secondary institution (i.e., college or university) to another. Students might choose to transfer for any number of reasons; common motivating factors include the opportunity to obtain both advanced training and multiple credentials in order to increase the number of future employment options. Additionally, students whose high school grades do not allow them to enter their program or institution of choice might instead enroll first in an institution with less stringent admission requirements. Obtainment of the initial post-secondary credential (e.g., diploma) can then facilitate transfer into the desired credential (e.g., degree) (Lang and Lopes, 2014), particularly when both are within related fields (e.g., Computer Programming diploma and Computer Science degree).

Transferring within similar fields of study often means that there is overlap in topics and/or courses required for both credentials; therefore, in order to effectively recognize students' previous learning, receiving institutions are often required to assess "transfer credit." Although numerous factors might influence this assessment, learning outcomes are considered a particularly valuable tool in the process (Arnold et al., 2020a). Learning outcomes are the measurable objectives defined at the end of an assignment, class, course, or program (Davis, 2009) and indicate the skill or knowledge level that can be expected from a student who has successfully completed the task in question. When a student transfers between institutions, the receiving institution typically reviews course learning outcomes from the previous institution to determine whether they align with the learning outcomes of comparable courses offered at the receiving institution. Generally, program coordinators or other domain experts (e.g., teaching faculty) are the trusted authority designated to determine whether credit is warranted; however, human evaluation can be a complex and challenging task (Fallon, 2015).

The process of assessing transfer credit can be facilitated through the use of Natural Language Processing (NLP) based semantic similarity algorithms. NLP has wide applicability, with the main challenge of measuring textual semantic similarity (Chandrasekaran and Mago, 2021b; Majumder et al., 2016). In the past few decades, there has been rich advancement in defining various measures for similarity between words, short texts, and sentences (Corley and Mihalcea, 2005; Ramage

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 38–47
September 11–12, 2023.

et al., 2009). Word-embeddings have emerged as a well-known technique that represents text in the form of a real-valued vector that reasonably captures the syntactic and semantic resemblance between them (Turian et al., 2010; Mikolov et al., 2013). Transformer-based pre-trained language models trained on large text corpora have successfully emerged to be paradigmatic models for building vector-based representations of texts (Vaswani et al., 2017). These models have applications in numerous fields such as text summarization (Mohamed and Oussalah, 2019), question/answering (Bordes et al., 2014; Lopez-Gazpio et al., 2017), sentiment analysis (Zhao et al., 2016), and sentence prediction, among others.

In this direction, this paper aims to propose two novel datasets consisting of course learning outcomes in postsecondary education. We determined the complexity of the outcomes (sentences) through readability analysis. We also implemented various embedding models to scrutinize the similarity between pairs of sentences and compared the models' performance with human-annotated results. Among different models, we found that ALBERT, MPNET and DistilRoBERTa demonstrated the best ability to predict the similarity between pairs of learning outcomes. However, Davinci - a GPT-3 model which is expected to predict better results - is only able to provide a good qualitative explanation and not an accurate similarity score.

2 Context and Motivation

2.1 Learning Outcomes in Postsecondary Education

Learning outcomes are "clearly defined and measurable statements of learning that reflect the scope and depth of performance; what a learner is expected to know, understand and be able to demonstrate after completion of a process of learning" (Lennon et al., 2014, p. 47). Within postsecondary education, outcomes are foundational for both developing curriculum and demonstrating quality assurance (Arnold et al., 2020a; Lennon, 2015). Transfer credit assessment increasingly relies on learning outcomes as a means of evaluating similarity between courses and credentials offered by different postsecondary institutions (Arnold et al., 2020a; Fallon, 2015), with outcomes sometimes being viewed as a "currency" that students can exchange between institutions in order to avoid repeating previous learning (Young et al., 2017). Effectively assessing transfer credit is an important process when considering that the amount of credit received can correspond to increase in academic performance as well as influence academic workload and time to completion for obtaining a postsecondary credential (Gerhardt and Masakure, 2016).

2.2 The Challenge of Assessing Learning Outcomes

Learning outcomes have the potential to establish a common language for communicating student learning and achievement across contexts (Arnold et al., 2020b); however, the overall process of assessing transfer credit tends to be both resource- and time-intensive (Arnold et al., 2020a). Additionally, course comparisons can differ substantially across institutions, and might (or might not) incorporate numerous other considerations related to content, evaluation, and grading (Arnold et al., 2020a). This subjectivity can be detrimental for students and institutions alike (Tortola et al., 2020), with the lack of consistency in standards and processes presenting a notable barrier. A recommendation to address this concern is the implementation of policies and practices that facilitate consistent decision-making, for example by documenting previous assessments (Wheelahan et al., 2016). An additional consideration is the presence of common assumptions regarding the nature and quality of education offered at different types of institutions (e.g., colleges and universities) (Arnold et al., 2020b), which could influence transfer credit decisions. Again, establishing some means of consistency that eliminates such potential bias could facilitate a more accurate and effective assessment process.

3 Methodology

3.1 New Dataset Development

We developed two novel datasets consisting of learning outcomes related to two content areas, namely (1) human anatomy and (2) operating systems. To create each dataset, we first accessed relevant course outlines from postsecondary institutions in Ontario, Canada. All of the outlines were publicly available and could be accessed via the institutions' websites without log-in credentials or other permissions. Next, we extracted the learning outcomes from each course outline and organized them by field (i.e., human anatomy and operating systems), institution (e.g., Institution A, Institution B, etc.), course (e.g., ANAT 101, BIOL 102, etc.), and topic (e.g., Digestive System, Muscular System, etc.). In some instances, we modified the general sentence structure of a learning outcome to either reduce“wordiness”, delete redundant information, and/or separate information pertaining to multiple topics. For example, an outcome that included two topic areas, such as “Explain the structure and function of the muscular and skeletal systems,” would become two separate outcomes (e.g., “Explain the structure and function of the muscular system; Explain the structure and function of the skeletal system”). The resulting datasets consisted of 28 (anatomy) and 59 (operating systems) unique learning outcomes (sentences) representing the knowledge and skills that would be expected of students who successfully completed the respective courses.

To create sentence pairs for analysis, learning outcomes from each dataset were paired together so that (1) both similar and dissimilar pairs were represented uniformly (i.e., by creating both inter- and intra-topic pairings) and (2) no learning outcomes were repeated more than twice. A total of 28 and 45 sentence pairs were analyzed for the anatomy and operating systems datasets, respectively. The datasets are available at https://github.com/JAkriti/New-Dataset-and-Performance-of-Embedding-Models.

3.2 Complex Sentence Dataset (Chandrasekaran and Mago, 2021a)

Recently, a dataset comprising 52 sentence pairs related to definitions of Computer Science terminology was developed and analyzed. The authors conduct readability analysis anticipating that their dataset exhibits a low readability index. This claims that their dataset is more complex in comparison to two benchmark datasets (Sentences Involving Compositional Knowledge “SICK”(Marelli et al., 2014) and Semantic Text Similarity “STS” (Shao, 2017)). Their main objective is to show how the increase in complexity of sentences leads to a significant decrease in the performance of embedding models.

3.3 Readability Analysis

The readability score is a metric defined to measure the complexity of a sentence and deliberate the grade level of education required for a person to understand the piece of text. Depending on the complexity of learning outcomes, it is important to comprehend how reasonably the embedding models perform to evaluate the similarity scores between them. The indices used to determine the readability scores of the sentences in the proposed datasets are – a) Flesch-Kincaid Grade Level (Coleman and Liau, 1975), b) Coleman-Liau Index (Kincaid et al., 1975), c) Automated readability Index (Kincaid et al., 1975), d) Linsear Write and e) Gunning fog index (Gunning et al., 1952).

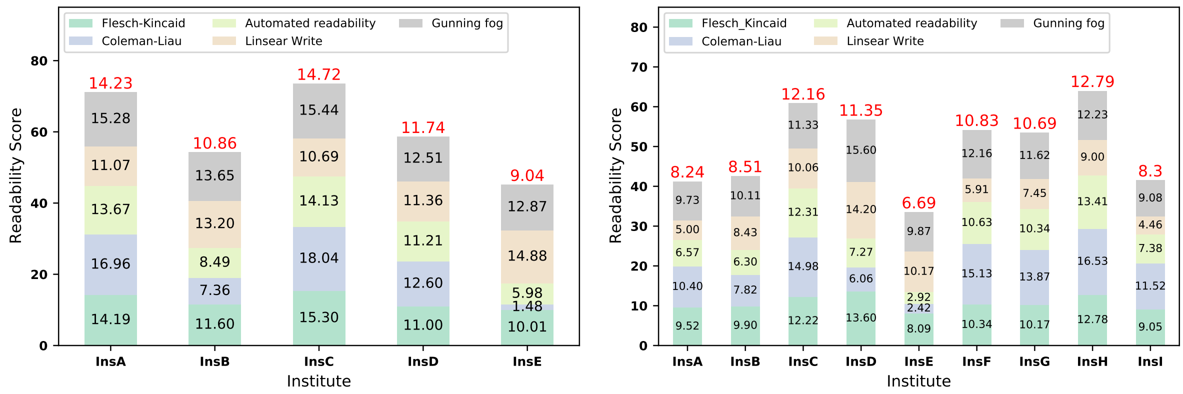
The readability scores of learning outcomes from each institute (e.g., Institute A, Institute B, etc.) are evaluated using the above indices. The aggregate of all these indices provides an overall readability score of each institute as highlighted in Figure 1 (for each dataset). For example, an average score of 11.74 shows that a reader needs a qualification of grade 11 to understand the text. Therefore, following this notation we observe that a reader requires education of collegiate level and above to understand the Anatomy sentences, and knowledge of grade 12 and above for Computer Science sentences.

3.4 Annotation

To develop a basis for comparing the performance of embedding models, the proposed datasets are each manually evaluated by three human respondents with relevant contextual expertise. The Anatomy dataset is evaluated by two graduated scholars and one graduate student in Kinesiology. The Computer Science dataset is evaluated by three thesis-based Master’s students. The annotators have been made aware of the applicability of this work. Each sentence pair is annotated on a scale of 0 to 9, where 0 (9) represents completely dissimilar (similar) sentences. To affirm the competency of these human ratings, we computed inter-rater agreement using *Krippendorff’s alpha coefficient* represented as α, where data with a coefficient value between $0.667 < \alpha < 0.8$ is considered reliable (Krippendorff, 2011; Hayes and Krippendorff, 2007). For the Anatomy dataset, $\alpha = 0.71$, and for the Computer Science dataset, $\alpha = 0.68$ which indicates that the annotation is reliable.

3.5 Web Interface

To ensure that the implementation of pre-tarined embedding models is successful in assisting transfer credit assessment, a web interface is developed to streamline the process. This begins by prompting users to upload new programs to the website



(a) Anatomy Dataset

(b) Computer Science Dataset

Figure 1: Readability analysis of learning outcomes from different institutes (denoted as InsA, InsB, and so on) for (a) Anatomy (b) Computer Science dataset using five different indices (values indicated in black). The aggregate scores are highlighted in red on each of the stacked bar graph.

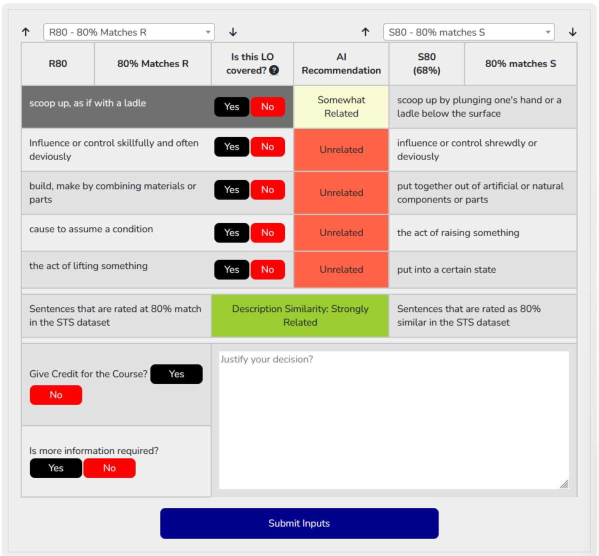


Figure 2: An example of the web interface where learning outcome comparisons can be observed

that contain information about the courses and their expected learning outcomes. Once an institute identifies a program they would like to transfer credit to, a comparative analysis is performed where a natural language processing algorithm is used to determine the semantic similarity between each course.

From these results, members of the receiving institute are able to access the screen shown in Figure 2 where they can observe suggestions from the algorithm for each learning outcome comparison before making their own decisions. After each user has provided input, the owner of the analysis can then observe the overall consensus before making a final recommendation on the transfer credit and generating a report to show the outcome.

Model	Version
$\text{BERT}_{\text{base}}$ (Devlin et al., 2018)	*bert-base-nli-mean-tokens*
$\text{BERT}_{\text{Large}}$ (Devlin et al., 2018)	*bert-large-nli-mean-tokens*
$\text{RoBERTa}_{\text{base}}$ (Liu et al., 2019)	*roberta-base-nli-mean-tokens*
$\text{RoBERTa}_{\text{Large}}$ (Liu et al., 2019)	*nli-roberta-large*
ALBERT (Lan et al., 2019)	*paraphrase-albert-small-v2*
DistilRoBERTa (Sanh et al., 2019)	*all-distilroberta-v1*
DistilRoBERTa (Sanh et al., 2019)	*nli-distilroberta-base-v2*
MPNeT (Song et al., 2020)	*all-mpnet-base-v2*
GPT-3 (Brown et al., 2020)	Davinci OpenAI

Table 1: Pre-trained embedding models used to generate sentence embeddings.

3.6 NLP Algorithm

Transformer is a neural network architecture that emerged as a breakthrough in NLP (Vaswani et al., 2017). Along with the encoder-decoder structure, self-attention mechanism is the key characteristic of transformers for the algorithms to learn the long-range relationship between words in a sequence. This architecture has surpassed the performance of various traditional networks like convolutional and recurrent neural networks known for language understanding (Mikolov et al., 2011). Furthermore, *Sentence transformer* is a transformer-based model designed to generate a fixed-size dense vector for a sentence of any length (Reimers and Gurevych, 2019). A brief outline of transformer-based models along with their sentence transformer version used in this paper is given in Table 1. The resulting sentence embeddings are then compared using cosine similarity.

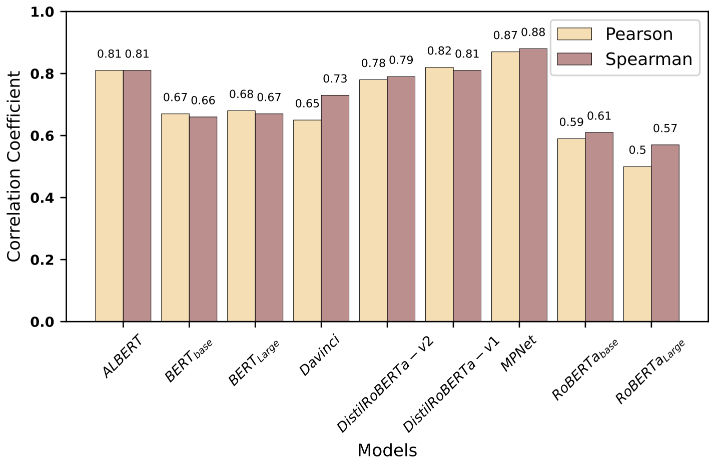


Figure 3: Pearson's and Spearman's correlation coefficient to analyse relationship between similarity values of human annotators and embedding models (Section 3.6) for Anatomy dataset.

S1: Discuss the structural organization and function of the respiratory system and its major organs.
S2: List the parts of the respiratory system and identify their functions.

ALBERT	$BERT_{base}$	$BERT_{Large}$	Davinci	DistilRoBERTa-v1	DistilRoBERTa-v2	MPNeT	$RoBERTa_{base}$	$RoBERTa_{Large}$	Human
0.8288	0.8197	0.9002	0.9098	0.7738	0.7967	0.7706	0.8901	0.8622	0.7407

Table 2: Similarity scores of a sentence pair from Anatomy dataset, evaluated using versions of pre-trained embedding models discussed in Section 3.6. Human ratings are normalized between 0 and 1.

4 Results

This section provides an extensive comparative analysis of various embedding techniques (discussed in Section 3.6) implemented for evaluating the similarity scores of learning outcomes in proposed datasets. To evaluate the relationship between the similarity scores of human annotators and embedding models we employ Pearson's and Spearman's rank coefficients for the Anatomy and Computer Science datasets (including the dataset proposed by (Chandrasekaran and Mago, 2021a) and the dataset proposed in this paper).

4.1 New Proposed datasets

4.1.1 Human Anatomy dataset

The results presented in Figure 3 demonstrate Pearson's and Spearman's correlation coefficients for the proposed Anatomy dataset of learning outcomes. While comparing different pre-trained embedding models we found that MPNeT achieves the best performance with 0.87 Pearson's value and 0.88 Spearman's value. This shows that the scores of MPNeT are very highly correlated to the human annotator's ratings. Moreover, DistilRoBERTa-v1 and ALBERT models show good performance with nearly 0.81 for both Pearson's and Spearman's correlation predicting that the ratings are highly correlated. However, every other model fails to predict a better similarity score. An example of sentence pair specifying how similar the sentences are according to embedding models is highlighted in Table 2, where we clearly observe that MPNeT is closest to human rating.

Example 1- ChatGPT

S1: Apply the basic planes of motion and fundamental movements associated with human movement using proper anatomical terminology.
S2: Describe the structure and function of bones and bony prominence.

There is not a strong similarity between the two sentences. S1 is about applying knowledge of human movement and anatomy, while S2 is about describing the structure and function of bones. They both involve anatomy, but they are not closely related in terms of content.

Davinci: 0.8201 | Human score: 0.6667

On the other hand, where Davinci model is anticipated to provide better results, we observed that though this model is not able to predict a reliable similarity score, it provides a piece of genuine qualitative information as shown above. This sentence pair is expected not to be similar as they are related to two different aspects of human anatomy. However, Davinci provides a high similarity score 0.8201 which is far away from the human similarity score, although it reflects a good explanation when explored on ChatGPT.

4.1.2 Computer Science dataset

For the Computer Science dataset proposed in this paper, the performance of embedding models is displayed in Figure 4. Also, an example of sentence pair with similarity values between them from different models is given in Table 3. Clearly, among the various embedding models, DistilRoBERTa-v2 shows the highest value for Pearson's and Spearman's coefficient displaying a very good correlation between the two ratings. However, we observed that the annotated values for this dataset fail to follow the normal distribution. Therefore, it is not reasonable to certainly rely on Pearson's test to judge the correlation between human evaluation and embedding models. Furthermore, Davinci- the most capable GPT-3 model fails to determine a good correlation. However, this model provides a justifiable interpretation of the similarity between the two sentences as shown in the example below.

Example 2- ChatGPT

S1- Design and install operating system to experience its functioning.
S2- Configure Package repositories and package management.

There is some similarity between the two sentences. Both S1 and S2 involve working with operating systems and installing or configuring different components or features. However, the specific tasks described in the sentences are not closely related. S1 involves designing and installing an operating system to experience its functioning, while S2 involves configuring package repositories and package management.

Davinci: 0.8192 | Human score: 0.0222

Here the explanation is true and acceptable, even though the Davinci similarity score is very high i.e. 0.8192 in comparison to human rating.

4.2 Complex Sentence dataset(Chandrasekaran and Mago, 2021a)

In this paper, the authors show how the embedding models do not perform well to predict the similarity values of complex sentence dataset proposed by them. We try to analyze these sentence pairs using the versions of models highlighted in Section 3.6. Notably, we found that ALBERT model executes best performance with 0.83 Pearson's coefficient and 0.77 Spearman's value as shown in Figure 5. Furthermore, DistilRoBERTa-v2, DistilRoBERTa-v1, and MPNeT have high Pearson's (Spearman's) value of 0.81(0.78), 0.81(0.78), and 0.79(0.77) respectively which outperforms the results of the models presented in (Chandrasekaran and Mago, 2021a). Table 4 shows the results of embedding models for a particular sentence pair from the dataset where ALBERT performs the best in comparison to human ratings. However, the Davinci model which is expected to predict better results is only able to provide a good explanation and not a valid score. For example, the subjective description of the similarity between a chosen sentence pair from the dataset is very well elaborated by ChatGPT, as shown below. Whereas, this model predicts a high similarity score of 0.7639 which shows the sentences have a good similarity which is not actually true as observed from the human score.

Example 3- ChatGPT

S1: A procedure for solving a mathematical problem in a finite number of steps that frequently involves repetition of an operation.
S2: Various methods or formats for organizing data in a computer

There is not a strong similarity between the two sentences. S1 is describing an algorithm, which is a specific type of procedure used to solve a mathematical problem. S2 is describing ways of organizing data in a computer. While both sentences involve some type of procedure or method, they are not closely related in terms of content.

Davinci: 0.7639 | Human score: 0.1200

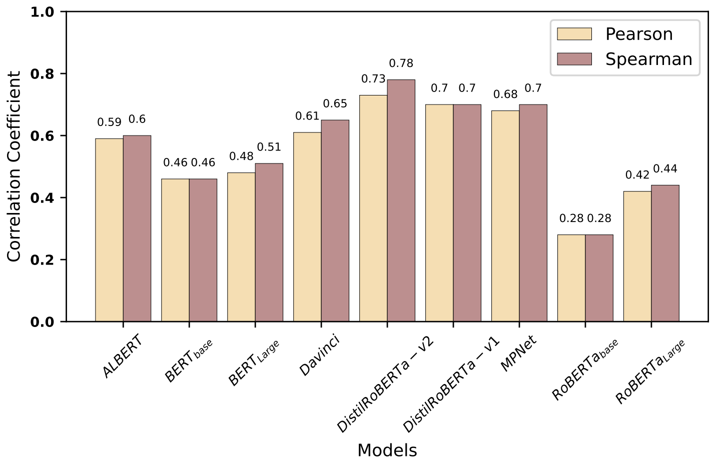


Figure 4: Pearson's and Spearman's correlation coefficient to analyse relationship between similarity values of human annotators and embedding models (Section 3.6) for proposed Computer Science dataset.

S1- Manage securely remote systems. S2- Maintain a Unix workstation and set it up as a network client.									
ALBERT	$BERT_{base}$	$BERT_{Large}$	Davinci	DistilRoBERTa-v1	DistilRoBERTa-v2	MPNeT	$RoBERTa_{base}$	$RoBERTa_{Large}$	Human
0.2695	0.5247	0.5274	0.8145	0.4241	0.5708	0.4246	0.4882	0.4653	0.6667

Table 3: Similarity scores of a sentence pair from proposed Computer Science dataset, evaluated using versions of pre-trained embedding models discussed in Section 3.6. Human ratings are normalized between 0 and 1.

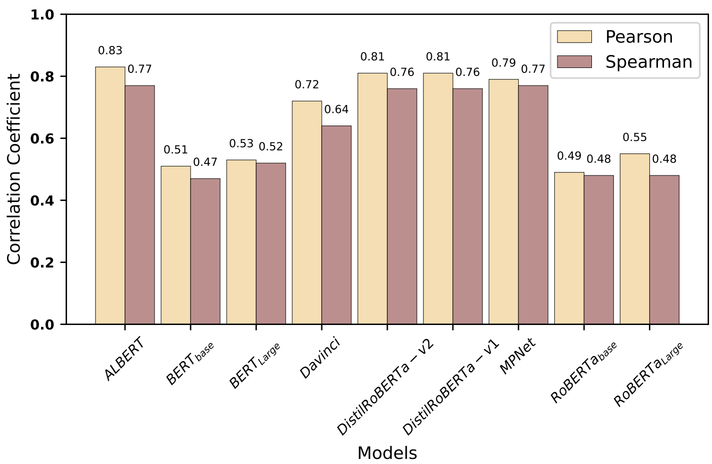


Figure 5: Pearson's and Spearman's correlation coefficient to analyse relationship between similarity values of human annotators and embedding models (Section 3.6) for complex sentence dataset.

S1- A procedure for solving a mathematical problem in a finite number of steps that frequently involves repetition of an operation. S2- Various methods or formats for organizing data in a computer.									
ALBERT	$BERT_{base}$	$BERT_{Large}$	Davinci	DistilRoBERTa-v1	DistilRoBERTa-v2	MPNeT	$RoBERTa_{base}$	$RoBERTa_{Large}$	Human
0.1300	0.4967	0.5656	0.7639	0.1623	0.3924	0.2001	0.6812	0.4069	0.0667

Table 4: Similarity scores of a sentence pair from Complex sentence dataset, evaluated using versions of pre-trained embedding models discussed in Section 3.6. Human ratings are normalized between 0 and 1.

5 Conclusion

Transfer credit assessment usually consists of course comparisons via the evaluation of learning outcomes, which represent an important tool for assessment but are also subject to potential inconsistencies and bias. Therefore, an automated system to assess transfer credit based on learning outcomes across institutes can facilitate the process by providing a reliable and consistent measure of similarity. Over the years there has been a rich advancement in the era of large language models to measure semantic similarity between texts. Various pre-trained embedding models have been developed to represent text for algorithms to understand and compare semantic similarity. In this paper, we aim to propose two novel datasets of learning outcomes for courses in Human Anatomy and Computer Science operating systems and perform an analysis using embedding models to assist in transfer credit assessment. We found that versions of ALBERT, MPNeT and DistilRoBERTa outperform Davinci (a GPT-3 model) that only provides a good qualitative interpretation of the similarity between pairs of sentences. Application of these models within the context of transfer credit assessment can contribute to greater efficiency and consistency when determining learning outcome similarity.

6 Limitations

Due to the complexity measures (readability analysis) requiring a minimum of 100 words, some of the smaller learning outcome sets require padding. To try and minimize the effect this will have on the results, we append the word "a" until the set can be measured. Furthermore, the datasets involve learning outcomes from the same courses being offered at different years of class. Therefore, while conducting human annotation, the comparison among learning outcomes is not consistent, which leads to a low inter-rater agreement among human values for both datasets. While utilizing the pre-trained embedding models, due to fewer number of sentences in the dataset we were not able to pre-train the models. This reflects a need to further enhance the dataset.

Acknowledgment

The authors would like to thank the Research Team, DaTALab, Lakehead University, for their support. They would also like to thank Lakehead University, CASES, and the Ontario Council for Articulation and Transfer, without their support this research would not have been possible.

References

Christine Arnold, Mary Wilson, Jean Bridge, and Mary Catherine Lennon. 2020a. Learning outcomes, academic credit and student mobility.

Christine Arnold, Mary Wilson, Michael Potter, and Leesa Wheelahan. 2020b. Shifting paradigms in postsecondary education: Historical, conceptual, and theoretical frameworks governing outcomes-based approaches to credit transfer. *Learning Outcomes, Academic Credit and Student Mobility*, 201:199.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. *arXiv preprint arXiv:1406.3676*.

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.

Dhivya Chandrasekaran and Vijay Mago. 2021a. Comparative analysis of word embeddings in assessing semantic similarity of complex sentences. *IEEE Access*, 9:166395–166408.

Dhivya Chandrasekaran and Vijay Mago. 2021b. Evolution of semantic similarity—a survey. *ACM Computing Surveys (CSUR)*, 54(2):1–37.

Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283.

Courtney D Corley and Rada Mihalcea. 2005. Measuring the semantic similarity of texts. In *Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment*, pages 13–18.

Barbara Gross Davis. 2009. *Tools for teaching*. John Wiley & Sons.

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*.

Nicole Fallon. 2015. Leaning outcomes in credit transfer: A key tool for innovation in student mobility.

Kris Gerhardt and Oliver Masakure. 2016. Postsecondary student mobility from college to university: Academic performance of students. *Canadian Journal of Higher Education*, 46(2):78–91.

Robert Gunning et al. 1952. Technique of clear writing.

Andrew F Hayes and Klaus Krippendorff. 2007. Answering the call for a standard reliability measure for coding data. *Communication methods and measures*, 1(1):77–89.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Daniel Lang and Valerie Lopes. 2014. Deciding to transfer: A study of college to university choice. *College Quarterly*, 17(3):1.

Mary Catharine Lennon. 2015. Incremental steps towards a competency-based post-secondary education system in ontario.

Mary Catherine Lennon, Brian Frank, James Humphreys, Rhonda Lenton, Kirsten Madsen, Abdelwahab Omri, and Roderick Turner. 2014. *Tuning: Identifying and measuring sector-based learning outcomes in postsecondary education*. Higher Education Quality Council of Ontario Toronto.

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*.

Iñigo Lopez-Gazpio, Montse Maritxalar, Aitor Gonzalez-Agirre, German Rigau, Larraitz Uria, and Eneko Agirre. 2017. Interpretable semantic textual similarity: Finding and explaining differences between sentences. *Knowledge-Based Systems*, 119:186–199.

Goutam Majumder, Partha Pakray, Alexander Gelbukh, and David Pinto. 2016. Semantic textual similarity methods, tools, and applications: A survey. *Computación y Sistemas*, 20(4):647–665.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223.

Tomáš Mikolov, Anoop Deoras, Daniel Povey, Lukáš Burget, and Jan Černockỳ. 2011. Strategies for training large scale neural network language models. In *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 196–201. IEEE.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751.

Muhidin Mohamed and Mourad Oussalah. 2019. Srl-esa-textsum: A text summarization approach based on semantic role labeling and explicit semantic analysis. *Information Processing & Management*, 56(4):1356–1372.

Daniel Ramage, Anna N Rafferty, and Christopher D Manning. 2009. Random walks for text semantic similarity. In *Proceedings of the 2009 workshop on graph-based methods for natural language processing (TextGraphs-4)*, pages 23–31.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Yang Shao. 2017. Hcti at semeval-2017 task 1: Use convolutional neural network to evaluate semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 130–133.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867.

Elisa Tortola, Christine Arnold, and Zanele Myles. 2020. Foundations for learning outcomes and credit transfer. *Learning Outcomes, Academic Credit and Student Mobility*, 201:19.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394.

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.

Leesa Wheelahan, Gavin Moodie, Mary Catherine Lennon, Amanda Brijmohan, and Eric Lavigne. 2016. Student mobility in ontario: A framework and decision making tool for building better pathways. *Centre for the Study of Canadian and International Higher Education, OISE-University of Toronto: Toronto, ON, Canada*.

Stacey Young, PG Piché, and Glen A Jones. 2017. Two towers of transformation: The compatibility of policy goals of differentiation and student mobility. *Toronto: Center for the Study of Canadian and International Higher Education, OISE-University of Toronto*.

Jun Zhao, Kang Liu, and Liheng Xu. 2016. Sentiment analysis: mining opinions, sentiments, and emotions.

Because is why: Children's acquisition of topoi through why questions

Christine Howes, Ellen Breitholtz and Vladislav Maraev
Department of Philosophy, Linguistics and Theory of Science
University of Gothenburg
christine.howes@gu.se, ellen.breitholtz@ling.gu.se, vladislav.maraev@gu.se

Abstract

In this paper we look at how children learn the underlying principles of commonsense reasoning, sometimes referred to as *topoi*, which are prevalent in everyday dialogue. By examining the utterances of two children in the CHILDES corpus for whom there is extensive longitudinal data, we show how children can elicit topoi from their parents by asking why-questions. This strategy for the rapid acquisition of topoi peaks at around age three, suggesting that it is a critical step in becoming a fully competent language user.

1 Introduction

Children pick up language with remarkable ease. From not being able to speak at all they learn in a few years to be fully competent language users. This does not just mean being able to communicate meaning coded in words and phrases – it also involves inference and association from the linguistic expressions and non-linguistic actions used to a meaning in use. Breitholtz (2020) discusses how such inferences draw on globally accepted facts ("the sun sets in the west"), norms ("one loves one's family"), and other principles of reasoning ("If you can do a and a is more difficult than b you can also do b"). Principles like these often implicitly underpin conversational moves, episodes in conversation and entire discourses, and have been claimed to be essential to capturing linguistic meaning in use (Ducrot, 1988; Anscombre, 1995). In their account Ducrot and Anscombre draw on Aristotelian dialectic and rhetoric, and use the term *topoi* (sg. *topos*) for such principles. Familiarity with the topoi that are acceptable in a community is also important for being proficient in a new language, as well as interpreting the behaviour of others. We see evidence of this in (1), discussed in Breitholtz and Howes (2020) where a father and son engage in a discussion about whether or not Lee, the son, could still play football even though he is not going to school because of illness. Both Dave and Lee are reasoning in a pragmatically competent way, despite evoking different topoi such as "one should rest when one is ill", "disease spreads less outdoors" (and possibly "fresh air is healthy") and "if one is well enough to do something less important and more exerting, one is also well enough to do something more important and less exerting".

(1) Dave: . . . you're gonna be home from football until four, you gonna have your dinner, want a bath.
Lee: Yeah, but I might not go to school tomorrow.
Dave: Why?
Lee: Cos of my cough.
Dave: How can you play football and not go to school then?
Lee: Cos I was going out in the fresh air, I'm alright when I'm out in the fresh air.
Dave: So why aren't you going to school then?
Lee: I'm in the class room all day dad.
[BNC KBE 10554-10561]

As this dialogue illustrates, a pre-teen child is capable of sophisticated argumentation drawing on principles which are also recognised by his adult discussion partner. However, the topoi Lee draws on in (1) have been learnt by him by explicit instruction, but also by via inference and induction. Breitholtz and Howes (2020) discuss how younger children, around four years of age, can be shown to have adopted topoi which they then generalise in non conventional ways. They also point out that one way for children to acquire topoi is through an extensive use of why-questions, and show that these peak at around 3 years (consistent with extensive evidence about children's stages of acquisition of wh-questions Bloom et al., 1982; Rowland

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 48–54
September 11–12, 2023.

et al., 2003; Valian and Casey, 2003). In this paper we probe this finding by looking at the longitudinal use of why-questions by two children in the CHILDES corpus for whom extensive longitudinal data is available.

2 Background

2.1 Reasoning in Dialogue

Reasoning is essential in communication since interacting with others frequently involves making non-logical common-sense inferences linking context, background knowledge and beliefs to utterances in the dialogue in order to understand one another. These underpinning principles of reasoning – referred to as topoi – have been discussed at length in the literature on rhetoric and argumentation (e.g. Toulmin, 2003, a.o.). However, the idea of rules of thumb available to language users, which justify statements, suggestions or other types of utterances goes back to dialectic and rhetoric. In modern times, the concept of topos was introduced in linguistics as a theory of linguistic meaning where parts of discourse are perceived as connected by topoi (Ducrot, 1988). On this view the topoi accessible to an individual do not constitute a monolithic logical system, but represents a set of resources at the disposal of a dialogue participant for producing and interpreting utterances and discourse contributions. Breitholtz (2020) shows how a theory of topoi relates to semantic-pragmatic theories such as Gricean implicature theory and Relevance theory (Grice, 1975; Sperber and Wilson, 1995), and how it can explain puzzles such as bridging inferences and certain types of discourse coherence (Clark, 1975; Asher and Lascarides, 2003).[1] Consider for example the exchange in (1), where Lee is trying to persuade his father Dave that he is well enough to play football but not well enough to go to school: In (1), both Lee and Dave base their argument on a generally accepted topos that being ill restricts certain activities, with Dave drawing on topoi about exertion, like "if you can do x and x is more exerting than y, you can also do y" – in fact a version of the "more and the less"-topos mentioned in the introduction – and Lee on other topoi having to do with the spread of disease and the health benefits of fresh air. In this dialogue sequence we see that an everyday conversation involves reasoning which cannot be accounted for using only traditional pragmatic theories where implicatures (Grice, 1975; Sperber and Wilson, 1995) are reached via assumptions of rationality and relevance. It also requires familiarity with a variety of topoi – principles about how it is acceptable to reason in different situations. Breitholtz and Howes (2020) suggest that topoi are learned through interaction with other agents and the world and show examples of where children draw on non-conventional topoi that they have learned by overextending inferences made in other instances of discourse. One such example is (2), where Greta, at 4 years and 3 months old in March 2020 demonstrates awareness of a topos related to the corona pandemic, namely that old people who contract the disease are more likely to die:

(2) Greta: What would happen if you drank the sea water?
Mother: It would make you poorly.
Greta: Really poorly?
Mother: Yes.
Greta: Old people would die. I don't know about us though.
[from Breitholtz and Howes (2020)]

In this example Greta overextends the topos that the elderly are more likely to die, if they contract coronavirus, to another situation where a young person would get ill.

2.2 Acquisition through interaction

Although traditional linguistics and developmental psychology started with the premise that there must be an innate language learning facility due to the presumed 'poverty of the stimulus' of a child's linguistic input (Berwick et al., 2011), there is a large body of evidence that refutes this position, from both a computational (Clark and Lappin, 2010) and a social perspective. This work (e.g. Halliday, 1975; Tomasello, 1992) emphasises the nature of language as action, and makes explicit the role of interaction in language acquisition. Specifically, research on child language acquisition underscores the importance of the social environment for the language learning child (Stephens and Matthews, 2014). Children are active in interactions with their caregivers long before they produce language and evidence suggests that it is this learning to interact (e.g. through gaze, Gredebäck et al., 2010; and turn-taking, Hilbrink et al., 2015; Casillas, 2014) which

[1] We should also note that, as pointed out by one of our reviewers, our approach theoretically and methodologically resonates with The Geneva Model of discourse analysis (see e.g. Filliettaz and Roulet, 2002).

bootstraps language acquisition (Levinson, 2006; Rączaszek-Leonardi et al., 2019). In a longitudinal study of the CHILDES-corpus (MacWhinney, 2000), Hiller and Fernández (2016) show that the type and amount of corrective feedback received by children affects their acquisition of particular grammatical phenomena. We hypothesise that adults' responses to why-questions and corrective feedback directed at topoi evoked by children will affect children's ability to seamlessly draw on topoi in conversation.

3 Why why?

Previous research shows that in adult conversation, topoi can be elicited by using why-questions (Schlöder et al., 2016). In many instances, what constitutes a good answer to a why-question constitutes an acceptable enthymeme when combined with the queried utterance, as is the case in (1), where "because of my cough" is an acceptable enthymematic reason for not going to school because of the acceptability of, for example, an underlying topos that when you are ill you should not mix with other people because you are contagious.

Where a dialogue participant cannot access or accommodate an appropriate topos, the asking of a why-question should be a particularly useful strategy to get one's interlocutor to make the topos more explicit. Indeed, when asked (fake) why-questions in a text-based dialogue experiment, people do provide the "missing" premises (Axelsson-Nord et al., 2021).

One reason that a dialogue participant may not have access to appropriate topoi is that there may be more than one applicable topoi available. However, in the case of young children it is often the case that a child lacks any topos that would make an argument coherent altogether.

We hypothesise that asking why-questions to increase the acceptable topoi one has access to is also a learning strategy for children, in line with evidence that children's why-questions are used for explanations and arguments (Bova and Arcidiacono, 2013)[2] once they have acquired a sufficient grounding in areas such as syntax (Cooper et al., 2023). Such a strategy – extrapolating and applying general principles of reasoning from minimal input (even when these go awry as in (2)) shows how children are capable of utilising informative learning signals to learn from limited data.

[2]It should be noted that why-questions can be used to express frustration, and not seek reasons in any real sense, but we leave this distinction to one side for future work. We thank one of our anonymous reveiwers for this point.

4 Method

For this exploratory study, we used two longitudinal cases from CHILDES. The specific sources and their characteristics are described in Table 1 (Henry, 1995; Rowland, 2007; Lieven et al., 2009). These were chosen based on the data collection being sufficiently fine-grained, and covering the proposed critical period for why-question acquisition at around age 3 (as shown in Figure 1 taken from Breitholtz and Howes, 2020).

Source	Description
Lara	Eng-UK/Lara; 120 recordings between age 1;9.13 – 3;3.25 (at home)
Thomas	Eng-UK/Thomas; 379 recordings between age 2;0.12 – 4;11.20 (mostly at home)

Table 1: Sources of data used

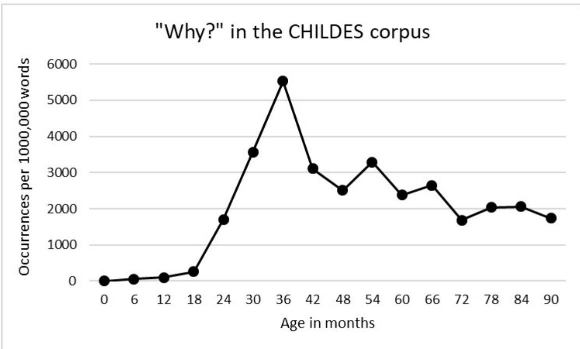


Figure 1: Frequency of 'why' in child language by age

We used PyLangAcq (Lee et al., 2016) to process the data and extracted all uses of 'why' split between those produced by the child and those produced by any other dialogue participant. While this will inevitably also pick up instances of 'why' which do not result in the giving of reasons (e.g. "I don't know why she did it") we believe it is a reasonable starting point for analysis with more fine-grained study left aside for future work. For comparison, we also extracted instances of 'because', once again split by child/any other dialogue participant. Because is often use to provide explanations (Eaton et al., 1999), and can thus also be analysed as making enthymemes more explicit in dialogue (as seen in example 1, where Lee respond

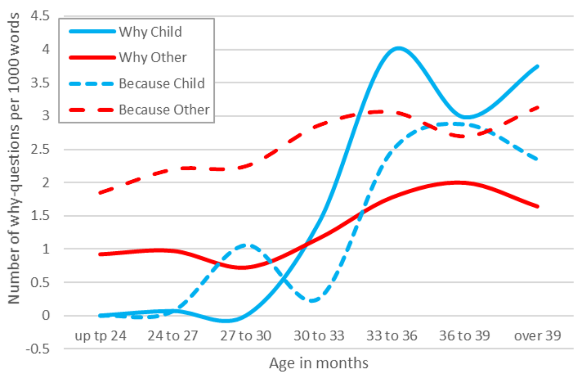


Figure 2: Lara: Frequency of 'why' and 'because' per 1000 words by age in months

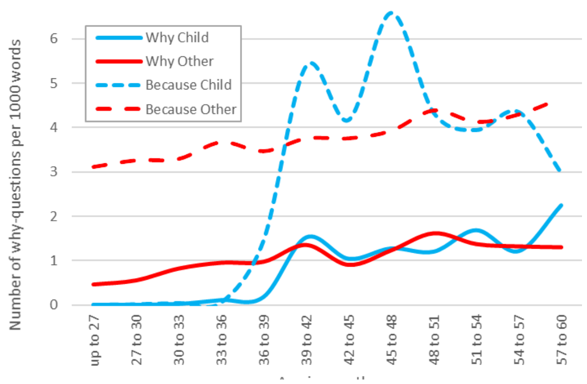


Figure 3: Thomas: Frequency of 'why' and 'because' per 1000 words by age in months

to his father's questions with because clauses that serve to illuminate the topoi that Lee is relying on in the dialogue).

5 Results & Discussion

As can be seen in Figures 2 and 3, both children have a peak of why-questions. For Lara, this occurs between 27 and 36 months whilst for Thomas this occurs between 36 and 42 months. Interestingly, while both show a distinct peak in why-questions (from asking none prior to this peak), this is relatively lower for Thomas, who peaks at approximately 1.5 why-questions per 1000 words, compared to Lara's 4. These differences in individual children are not apparent in the data shown in Figure 1. In this regard it is also informative to consider the input each child received in terms of why-questions, since acquisition of wh-questions in general has been linked to the input from the caregiver(s) (Rowland et al., 2003). While we cannot, of course, extrapolate from the available data to the total exposure of each child to why-questions, it is notable that in the available data, Thomas is also exposed to fewer why-questions than Lara, though the general patterns of why-questions they encounter is similar in both cases, rising steadily as the child asks more why-questions themselves. More fine-grained analysis is necessary to see how and whether these apparent contingencies have a direct impact on the child's interactive behaviour.

In terms of the use of because, once again we see that both children have a peak at around the same age range. Interestingly though, Thomas' peak in the use of 'because' coincides with his peak in the use of 'why', but is greater (in the order of 5-6 words per 1000). This corresponds to the relatively greater input of 'because' from other speakers that we see in the Thomas data, as compared to Lara.

Interestingly, these children seem to have potentially different strategies for acquiring topoi, with the differences not fully explainable by broad differences in input that we have looked at here.

5.1 Qualitative results

We now turn to some examples from the Lara corpus to illustrate how why-questions can elicit topoi. Example (3) is an early example of a why-question from Lara, which does elicit a topos regarding what types of behaviour are naughty. Note that in this case a similar question to an adult might have instead been answered by providing some motivation for Peter Rabbit's naughtiness, rather than the topos supplied here. We hypothesise that this is because of the expectation that competent adult users of the language will already have access to a topos which licences "stealing (lettuces from Mr McGregor's garden) is naughty", so the question would in fact be interpreted differently if the asker were an adult. This suggests that people interacting with small children who lack some rhetorical resources, are sensitive to this fact (even if this is not a conscious awareness), but this is an empirical question which future work should try to investigate.

(3) CHI: is that rabby all by himself ?
MOT: yes . he's in Mr McGregor's garden . he's naughty . [. . .]
CHI: is he naughty ?
MOT: yes .

CHI: why ?
MOT: why what ?
CHI: why he naughty ?
MOT: because he's gone into Mr McGregor's garden and he's stealing all his carrots and that's naughty , isn't it ?
[Lara 2;08;02]

Examples (4) and (5) are examples of the child asking more than one why-question in a row. This behaviour is very familiar to parents of children of around three, and we suggest that the initial answer may not satisfy the child's desire to access or accommodate an appropriate topos. Further work is needed to see how common such chains of why-questions are and whether these also occur at a critical age point or around the acquisition of particularly complex topoi.

(4) ELS: watch you don't break them now , Lara darling .
ELS: cause
CHI: why ?
ELS: cause Auntie Linda bought me them .
ELS: because I hafta look after them .
ELS: be very careful with them .
ELS: that's a good girl .
CHI: why ?
ELS: because I like them .
ELS: they're my special things .
CHI: them your special things ?
ELS: yes . *[Lara 2;10;14]*

Example (5) is also interesting as the topos it conveys is a normative one that in this particular family there is a 'rule' that one does not open a new treat if you already have one open. This may be a common rule in families, but there may also be differences between how children acquire such normative principles as opposed to, for example, globally accepted facts (for example, children learn that if you drop something it falls to the ground in the preverbal period). It is also not clear that children make this distinction at around the age they are producing a lot of why-questions. In example 6, Lara produces an enthymematic utterance which is underpinned by a normative topos which her mother rejects (that you have to have gloves on if you're gardening). In this case, Lara's why-questions seem to be targeting finding out what it is about this situation which means the normative topos that she has previously acquired (when you do gardening you wear gloves) does not apply.

(5) DAD: you're not opening that one until you've eaten all that one .
DAD: that's the
CHI: why ?
DAD: that's the rules , isn't it ?
CHI: why ?
DAD: er if you go in there and open it in there you're gonna be in big trouble .
MOT: we'll take it away from you .
DAD: you won't eat it .
CHI: pardon me ?
DAD: did you hear what I said ?
MOT: if you open it mummy will eat it .
DAD: do you hear what I'm saying , sugar ?
CHI: yes . *[Lara 2;10;14]*

(6) MOT: I was looking at what else we could we could plant in the garden .
CHI: you've gotta have gloves on .
CHI: but I haven't got any
MOT: you don't hafta have gloves on .
CHI: why ?
MOT: well .
MOT: you don't hafta .
CHI: why ?
MOT: well .
MOT: it's only (be)cause your hand get dirty .
[Lara 3;2;11]

6 Conclusions

As we have shown, children tend to have a peak of why-questions at around 3 years of age, which we speculate is due to their rapid acquisition of topoi at around this age. The development of the two children we have looked at in this paper is consistent with this. They also exhibit a peak in the use of 'because', although even in our small sample, the ways in which they use the available resources differs between the children suggesting there may be different pathways to acquiring topoi. One hypothesis is that if children are exposed to more explicit topoi (in the form of 'because' explanations) they may not have such a necessity to ask explicit why-questions. Further exploration of children's use of linguistic and pragmatic markers and their relationship to the interactive input is necessary to further elucidate these issues.

One of our plans for future research is to look at if and how 'why' and 'because' are complementary (and to what degree). These are intuitively codependent strategies (if you ask me why, I might expect a because), but how productive these strategies are has, to the best of our knowledge, not previously been investigated – particularly in child language data. More fine-grained analysis is necessary to investigate whether there are other aspects of the children's pragmatic acquisition strategy that co-varies with these two linguistic markers, but we leave such analysis to future work.

Although our data does not conclusively say that children whose parents use more 'because' explanations do the same, they suggest a connection between child behaviour and the behaviour of care givers in this respect. However, further work is needed to look at the relations between the frequency of why-questions and because-clauses in the language produced by caregivers and children.

Example (6) also suggests avenues for future work, since it indicates that the child has already acquired a topos and is now concerned with how far this topos can be generalised (in this case, the child has learnt that one usually wears gloves when gardening to keep one's hands clean, but that this is not necessary). Learning the scope and range of topoi is a critical –and non-trivial– task for the language learning child, as demonstrated by Greta's overextention of the topos in (2). It is noteworthy in this regard that young children are able to pick up and modify the topoi they have access to from very little input – something that is still beyond the capabilities of conversational AI.

Acknowledgements

This work was supported by an ERC Starting Grant, *DivCon: Divergence and convergence in dialogue: The dynamic management of mismatches* (10977927) and a Swedish Research Council Grant for the establishment of the *Centre for Linguistic Theory and Studies in Probability (CLASP)* (2014-0039).

References

Jean-Claude Anscombre. 1995. La théorie des topoi: Sémantique ou rhétorique? *Hermés*, 15.

Nicholas Asher and Alex Lascarides. 2003. *Logics of Conversation*. Cambridge University Press.

Ebba Axelsson-Nord, Vladislav Maraev, Ellen Breitholtz, and Christine Howes. 2021. What do you mean? Eliciting enthymemes in text-based dialogue. In *Proceedings of the 25th Workshop on the Semantics and Pragmatics of Dialogue - Poster Abstracts*. SEMDIAL.

Robert C. Berwick, Paul Pietroski, Beracah Yankama, and Noam Chomsky. 2011. Poverty of the stimulus revisited. *Cognitive Science*, 35(7):1207–1242.

Lois Bloom, Susan Merkin, and Janet Wootten. 1982. "wh"-questions: Linguistic factors that contribute to the sequence of acquisition. *Child development*, pages 1084–1092.

Antonio Bova and Francesco Arcidiacono. 2013. Investigating children's why-questions: A study comparing argumentative and explanatory function. *Discourse Studies*, 15(6):713–734.

Ellen Breitholtz. 2020. *Enthymemes and Topoi in Dialogue: The Use of Common Sense Reasoning in Conversation*. Brill, Leiden.

Ellen Breitholtz and Christine Howes. 2020. Communicable reasons: How children learn topoi through dialogue. In *Proceedings of the 24th Workshop on the Semantics and Pragmatics of Dialogue (WatchDial)*.

Marisa Casillas. 2014. Turn-taking. In Danielle Matthews, editor, *Pragmatic development in first language acquisition*, pages 53–70. John Benjamins Amsterdam, The Netherlands.

Alexander Clark and Shalom Lappin. 2010. *Linguistic Nativism and the Poverty of the Stimulus*. John Wiley & Sons.

Herbert H. Clark. 1975. Bridging. In *TINLAP '75 Proceedings of the 1975 workshop on Theoretical issues in natural language processing*, pages 169–174. Association for Computational Linguistics.

Robin Cooper, Jonathan Ginzburg, and Staffan Larsson. 2023. Learning language games probabilistically: From crying to compositionality. In J-P Bernardy, R Blanck, S Chatzikyriakidis, S Lappin, and A Maskharashvili, editors, *Probabilistic Approaches to Linguistic Theory*, pages 361–422. CSLI Publications.

Oswald Ducrot. 1988. Topoï et formes topiques. *Bulletin d'Etudes de Linguistique Francaise*, 22:1–14.

Judy H Eaton, Glyn M Collis, and Vicky A Lewis. 1999. Evaluative explanations in children's narratives of a video sequence without dialogue. *Journal of Child Language*, 26(3):699–720.

Laurent Filliettaz and Eddy Roulet. 2002. The geneva model of discourse analysis: an interactionist and modular approach to discourse organization. *Discourse Studies*, 4(3):369–393.

Gustaf Gredebäck, Linn Fikke, and Annika Melinder. 2010. The development of joint visual attention: A longitudinal study of gaze following during interactions with mothers and strangers. *Developmental science*, 13(6):839–848.

H.P. Grice. 1975. Logic and Conversation. *Syntax and Semantics*, 3(S 41):58.

Michael Alexander Kirkwood Halliday. 1975. Learning how to mean. In *Foundations of language development*, pages 239–265. Elsevier, Amsterdam.

Alison Henry. 1995. *Belfast English and Standard English: Dialect variation and parameter setting*. Oxford University Press on Demand.

Elma E. Hilbrink, Merideth Gattis, and Stephen C. Levinson. 2015. Early developmental changes in the timing of turn-taking: A longitudinal study of mother–infant interaction. *Frontiers in psychology*, 6:1492.

Sarah Hiller and Raquel Fernández. 2016. A data-driven investigation of corrective feedback on subject omission errors in first language acquisition. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 105–114.

Jackson L. Lee, Ross Burkholder, Gallagher B. Flinn, and Emily R. Coppess. 2016. Working with chat transcripts in python. Technical Report TR-2016-02, Department of Computer Science, University of Chicago.

Stephen C. Levinson. 2006. On the human "interaction engine'. In N. J. Enfield and Levinson S. C., editors, *Roots of Human Sociality: Culture, Cognition and Interaction*. The Wenner-Gren Foundation, Oxford.

Elena Lieven, Dorothé Salomo, and Michael Tomasello. 2009. Two-year-old children's production of multiword utterances: A usage-based analysis. *Cognitive Linguistics*.

Brian MacWhinney. 2000. *The CHILDES project: Tools for analyzing talk: Volume I: Transcription format and programs, volume II: The database*. MIT Press, Cambridge, MA.

Joanna Rączaszek-Leonardi, Nicole Rossmanith, Iris Nomikou, and Katharina J Rohlfing. 2019. Levels of coordination in early semantic development. *Psychology of Language and Communication*, 23(1):212–237.

Caroline F Rowland. 2007. Explaining errors in children's questions. *Cognition*, 104(1):106–134.

Caroline F Rowland, Julian M Pine, Elena VM Lieven, and Anna L Theakston. 2003. Determinants of acquisition order in wh-questions: Re-evaluating the role of caregiver speech. *Journal of child language*, 30(3):609–635.

Julian Schlöder, Ellen Breitholtz, and Raquel Fernández. 2016. Why? In *Proceedings of JerSem*, pages 5–14.

Dan Sperber and Deirdre Wilson. 1995. *Relevance: Communication and Cognition*, second edition. Blackwell, Oxford.

Gemma Stephens and Danielle Matthews. 2014. The communicative infant from 0-18 months: The social-cognitive foundations of pragmatic development. In Danielle Matthews, editor, *Pragmatic Development in First Language Acquisition*, pages 13–36. John Benjamins, Amsterdam.

Michael Tomasello. 1992. The social bases of language acquisition. *Social development*, 1(1):67–87.

Stephen E Toulmin. 2003. *The Uses of Argument*. Cambridge University Press, Cambridge.

Virginia Valian and Lyman Casey. 2003. Young children's acquisition of wh-questions: The role of structured input. *Journal of child language*, 30(1):117–143.

Do Language Models discriminate between relatives and pseudorelatives?

Adèle Hénot-Mortier
Department of Linguistics & Philosophy, Massachusetts Institute of Technology
32 Vassar Street, Cambridge, MA 02139, USA
mortier@mit.edu

Abstract

Large Language Models (LLMs) are often evaluated against massive benchmarks based on general-purpose tasks, which, despite being useful for concrete applications, tell us very little about the capacity of LLMs to learn *specific* and *challenging* aspects of the grammar. Here, we evaluate whether LLMs learn to identify a particular structure attested in Romance (and French in particular), called the pseudorelative. This structure, which is often surface-similar to a relative clause, is linked to robust syntactic and semantic restrictions. We present a series of experiments to test if LLMs pretrained on massive yet general corpora, manage to learn those various restrictions. Our results suggest that LLMs learn some but not all of these properties, but crucially fail at recognizing the most specific of them: cliticization.

1 Background on pseudorelatives

Pseudorelatives (**PR**s) (Schwarze (1974); Radford (1975); Kayne (1975); Guasti (1988) a.o.) resemble relative clauses (**RC**s) but exhibit a specific cluster of properties: (1) their head noun can be cliticized; (2) they only feature subject gaps; (3) they only appear below perception verbs; (4) they require the matrix and embedded tenses to match; (5) they imply the existence/truth of the embedded event even under matrix negation (Moulton and Grillo, 2015). Those various properties are illustrated below.

(1) Head noun cliticization
Jean **la** voit qui sourit.
Jean **3.SG.CL** sees that smiles.
'Jean sees her smiling.'

(2) Object gap (+cliticization)
* Jean la voit que Marc salue __.
Jean 3.SG.CL sees that Marc greets.
Intended: 'Jean sees Marc greeting her.'

(3) Non-perception verb (+cliticization)
* Jean la **pense** qui sourit.
Jean 3.SG.CL **thinks** that smiles.
Intended: 'Jean thinks she is smiling.'

(4) Tense mismatch (+cliticization)
* Jean la voit qui souri**ait**.
Jean 3.SG.CL sees.PRS that smiled.**PST**.
Intended: 'Jean sees her while she smiled.'

(5) Event presupposition
Jean ne la voit pas qui sourit.
Jean NEG 3.SG.CL sees NEG that smiles.
'Jean doesn't see her smiling (she **does**).'

Cliticization is perhaps the most robust diagnostic used to disambiguate PRs from RCs; without a cliticized head, and assuming conditions (2)-(5) are met, a PR will usually remain ambiguous with a (string-identical) RC. This is shown in (6).

(6) No cliticization: ambiguous parse
Jean voit Marie qui sourit.
Jean sees Marie that smiles.
'Jean sees Marie, who smiles.' (**relative**).
'Jean sees Marie smiling.' (**pseudorelative**)

Because of is rareness in corpora, its ambiguity with relative clauses, and the inability of LLMs to access external disambiguating cues (comma, intonation), the pseudorelative remains relatively opaque to current NLP benchmarks (Wang et al., 2018, 2019; Bowman et al., 2015; Williams et al., 2018; Rajpurkar et al., 2016, 2018; Zellers et al., 2018) which are used to assess LLMs' performances. Do LLMs pretrained on massive (but also very general, non-targeted, and therefore impoverished) corpora learn something about pseudorelatives anyway?

2 Preliminary corpus study

We run a corpus study to verify the claim that LLMs are mostly exposed to structurally ambiguous sentences such as (6). We start with simple exact Google queries following the patterns in (7), where V denotes one of the verbs listed in Table 1, and CL is a clitic pronoun (*le* or *la* if V starts with a consonant, *l'* if V starts with a vowel).

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 55–61
September 11–12, 2023. ©2023 Association for Computational Linguistics

(7) a. "Il V * qui"
He V wildcard that

b. "Il {le, la, l'} V qui"
He CL V that

The number of hits for these queries are gathered in Table 1. If some perception verbs are clearly more frequent than others (compare *voir*, 'see' vs. *épier*, 'spy on'), the tendency regarding cliticized constructions is clear: they are between 10,000 and 100,000 times less frequent than the string-ambiguous structures similar to (6).

exact query → V↓	(7a)	(7b)	$\frac{\#(7b)}{\#(7a)+\#(7b)}$
voit (see)	262,000,000	22,440	8.5e-5
apercoit (spot)	11,700,000	1,230	1.1e-4
regarde (look at)	192,000,000	6,370	3.3e-5
observe (watch)	51,100,000	759	1.5e-5
épie (spy on)	237,000	1	4.2e-6
surprend (catch)	21,900,000	247	1.1e-5
entend (hear)	70,200,000	7,820	1.1e-4
écoute (listen to)	121,000,000	18,200	1.5e-4

Table 1: Number of results for non-cliticized (ambiguous) and cliticized (unambiguous) PR structures returned by Google Search for different perception verbs.

To confirm this intuition, we matched a series of regular expressions[1] against a subset of the French OSCAR corpus (Ortiz Suárez et al., 2019; Caswell et al., 2021; Abadji et al., 2021), used to train models such as CamemBERT (Martin et al., 2020). The results shown in Table 2 confirm that a typical French LLM is mostly trained on ambiguous PR structures. Learning properties (1)-(5) would therefore require the models to exploit weak signals in the data to draw syntactic and semantic generalizations. The experiments that follow test whether LLMs achieve this goal – or not.

3 Experiment 1

Adapting a recent psycholinguistic experiment (Pozniak et al., 2019), we test if 8 LLMs trained on general French corpora (see Tab. 3 rows 1-8), learned the association between properties (3)-(4), pertaining to the type of the embedding verb and tense anaphoricity. The expected effects are: a

[1]The regular expressions were refined from the templates in (7) to include all possible subject pronouns and allowed up to 3 unspecified words in the wildcard (*). This restricts the search space for ambiguous relative constructions of the from of (6) but ensures that other constructions (such as an unambiguous relative clause located "far away" from the perception verb) are not matched by accident. It also allows to speed-up the search. Consequently, the matches and the proportions gathered respectively in the second and last columns of Table 2 should be respectively read as lower- and upper-bounds.

regular expression → V↓	(7a)'	(7b)'	$\frac{\#(7b)'}{\#(7a)'+\#(7b)'}$
voir	15157	168	1.1e-2
apercevoir	725	1	1.4e-3
regarder	2442	28	1.1e-2
observer	813	0	0.0
épier	13	0	0.0
surprendre	99	0	0.0
entendre	1975	27	1.3e-2
écouter	632	1	1.6e-3

Table 2: Number of matches for non-cliticized (ambiguous) and cliticized (unambiguous) regular expressions on 10,160,000 documents from the OSCAR corpus (containing a total of 52,037,098 documents).

preference for embedding of (pseudo)relatives under perception verbs (as opposed to e.g. stative verbs); a preference for matching tenses between the matrix clause and the embedded clause; an interaction between those two factors, favoring tense matching specifically under perception verbs. We take the interaction to be the most critical effect. These predictions were assessed reusing the 2×2 design (`verb_type`×`tense_match`) introduced by the original study. Example stimuli illustrating this design are given in (8) and their parameters are summarized in Table 4.

ID	Model	Lang.	Reference
1	flaubert_base_uncased	fr	Le et al. (2020)
2	camembert-base	fr	Martin et al. (2020)
3	gpt2-base-french	fr	(Cla)
4	gpt2-wechsel-french	fr	Minixhofer et al. (2022)
5	bert-base-multi-lingual-cased	multi	Devlin et al. (2018)
6	xlm-roberta-base	multi	Conneau et al. (2019)
7	xlm-roberta-large	multi	Conneau et al. (2019)
8	xlm-mlm-17-1280	multi	Lample and Conneau (2019)
9	bert-large-cased	en	Devlin et al. (2018)
10	gpt2-large	en	Radford et al. (2019)
11	xlnet-large-cased	en	Yang et al. (2019)

Table 3: Models used in Exp. 1 and 2

(8) Example stimuli reused from (Pozniak et al., 2019).

a. Marie a écouté le ministre qui critiquait le président.
b. ?Marie écoute le ministre qui critiquait le président.
c. Marie a été mariée au ministre qui critiquait le président.
d. Marie est mariée au ministre qui critiquait le président.

Sentence	`verb_type`	`tense_match`
(a.)	perception	y
(b.)	perception	n
(c.)	stative	y
(d.)	stative	n

Table 4: Summary of the 2×2 design of (Pozniak et al., 2019) reused in Exp. 1

Building on (Hale, 2001; Levy, 2008), our proxy for grammaticality was taken to be the log-probability assigned to a given sentence by the

LLM (see equations below). It was computed using the `minicons` library (Misra, 2022).

$$\textsc{Grammaticality}(w_t) \simeq -\textsc{Surprisal}(w_t)$$
$$= \log P(w_t|w_1 \dots w_{t-1})$$ [2]
$$\textsc{Grammaticality}(w_1 \dots w_t) \simeq -\sum_{i=1}^{t} \textsc{Surprisal}(w_i)$$

Linear mixed-effect modeling (performed with `statsmodels`, (Seabold and Perktold, 2010)) reveals that 6/8 LMs favor matching tenses, and 4/8 more so under perception verbs (`verb*tense` interaction) – supporting the expected interaction between (3) and (4) in French. Among the best performing models are a French-only (autoregressive) GPT-2 model (model 3) and a (bidirectional) multilingual RoBERTa model (model 7).

ID	best AIC?	verb_type	tense	interaction
1	n	. ✗	n.s.	. ✓
2	n	. ✓	** ✓	n.s.
3	y	n.s.	** ✓	* ✓
4	y	n.s.	** ✓	. ✓
5	n	n.s.	** ✓	n.s.
6	y	n.s.	** ✓	. ✓
7	y	n.s.	** ✓	* ✓
8	n	** ✓	n.s.	n.s.

Table 5: Significance results of LME modeling for `grammaticality` $\sim$ `verb_type`+`tense`+`verb_type`$*$ `tense` + (1|`frame`), where `frame` refers to the lexical skeleton shared by all stimuli in e.g. (8).[3]

English models (cf Tab. 3, rows 9-11) tested on English equivalents of the stimuli exemplified in (8), did not exhibit similar effects – consistent with English not allowing pseudorelatives. Plots of the distributions of grammaticality scores obtained with `xlm-roberta-large` (model 7) in both languages are given in Figure 1.

4 Experiment 2

We test the same LLMs on 4800 semi-automatically generated sentences following the template in (9) and differing in (1) head noun cliticization; (2) the gap's position (subject/object) and (3) the matrix verb's type (perception vs. attitude/action).

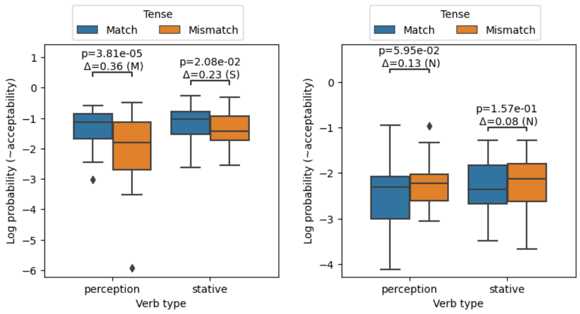
[2]In the case of BERT-like bidirectional models, this formula is adapted to masked language modeling: the probability of a word is computed given its left *and* right context.

[3]The 'best AIC?' column specifies if the formula yielded the lowest Akaike Information Criterion, as opposed to other simpler formulas without interactions or main effects. Other notations: '.' $= p \in].05; .1]$, '*'$= p \in].01; .05]$; '**'$= p \in [0; .01]$; ✓=coefficient validates the hypothesis; ✗=coefficient disproves the hypothesis.

[4]The scores are overall negative because they correspond to negative log probabilities (cf. equations above).

ID	best AIC?	verb type	tense	interaction
5	y	** ✓	* ✗	* ✗
6	n	. ✓	. ✗	. ✗
7	n	n.s.	* ✗	n.s.
8	n	** ✓	n.s.	n.s.
9	n	n.s.	n.s.	. ✓
10	n	** ✓	* ✓	n.s.
11	n	n.s.	n.s.	n.s.

Table 6: Significance results of LME modeling with English data. Same notations and parameters as Table 5. Strikingly, all but 1 model did not yield the best AIC for the formula involving an interaction term.



(a) French (b) English

Figure 1: Distributions of the grammaticality scores[4] for Exp. 1 with `xlm-roberta-large`. Δ refers to Cliff's Delta (non-parametric measure of effect size). N, S, M resp. mean 'negligible', 'small', 'medium'.

(9) Template for the stimuli of Exp. 2

$$\underset{\text{PRO}}{\left\{ \text{Il/Elle} \right\}} - \underset{\text{(CL)}}{\left\{ \begin{array}{c}\text{le/la/l'}\\ \emptyset\end{array} \right\}} - \underset{\text{V}}{\left\{ \begin{array}{c}\text{voit/...}\\ \text{pense/...}\end{array} \right\}} -$$

$$\underset{\text{(N)}}{\left\{ \begin{array}{c}\emptyset\\ \text{Marie/Jean}\end{array} \right\}} - \underset{\text{CP}}{\left\{ \begin{array}{c}\text{subject-gap relative}\\ \text{object-gap relative}\end{array} \right\}}$$

(10) Example stimuli for Exp. 2

a. Il voit Marie qui embrasse Jean.
b. Il voit Jean que Marie embrasse.
c. Il la voit qui embrasse Jean.
d. * Il le voit que Marie embrasse.
e. * Il pense Marie qui embrasse Jean.
f. * Il pense Jean que Marie embrasse.
g. * Il la pense qui embrasse Jean.
h. * Il le pense que Marie embrasse.

Given this design, we expect an overall preference for matrix perception verbs, subject gaps and non-cliticized constructions, but also a positive interaction between perception verbs and clitics, perception verbs and subject gaps, clitics and subject gaps, and all three variables together. As Tab.

Sentence	`clitic?`	`gap`	`verb_type`
(10a)	n	S	perception
(10b)	n	O	perception
(10c)	y	S	perception
(10d)	y	O	perception
(10e)	n	S	attitude
(10f)	n	O	attitude
(10g)	y	S	attitude
(10h)	y	O	attitude

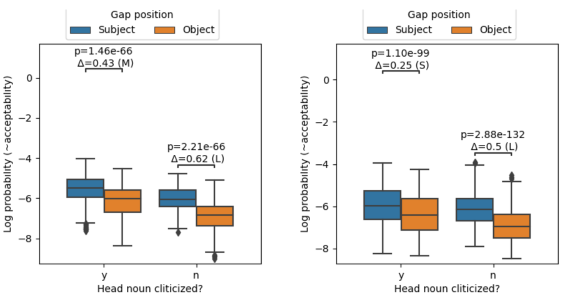
Table 7: Summary of the 2×2×2 design of Exp. 2

8 shows, linear mixed-effect modeling reveals a robust preference for subject-gaps (8/8 models, cf. col. 3) and more so under perception verbs (5/8 models, cf. col. 6), supporting (2)+(3). The desired `clitic*gap*verb_type` interaction however, was only captured by 1/8 models (cf. col. 8). Strikingly also, the interaction between cliticization and subject gaps is predicted by most models to have a negative effect on grammaticality, *contra* (1)+(2).

ID	v	g	c	v*c	v*g	c*g	v*c*g
1	. ✓	** ✓	** ✓	** ✗	. ✗	** ✓	n.s.
2	. ✓	** ✓	** ✓	** ✗	** ✓	** ✗	n.s.
3	n.s.	** ✓	** ✗	** ✓	** ✓	** ✗	. ✓
4	n.s.	** ✓	** ✗	** ✗	** ✗	** ✗	** ✓
5	n.s.	** ✓	** ✗	** ✓	n.s.	** ✗	n.s.
6	n.s.	** ✓	** ✗	** ✓	** ✓	** ✗	. ✗
7	n.s.	** ✓	** ✗	** ✓	** ✓	** ✗	** ✗
8	n.s.	** ✓	* ✗	** ✗	** ✓	**	n.s.

Table 8: Significance results of LME modeling for `grammaticality` $\sim$ `verb_type` + `gap` + `clitic` + `verb_type` $*$ `clitic` $*$ `gap`. Same notations as before.

The best performing model for this experiment appears to be a French-only GPT-2 model (model 3) – which was also among the best models for Exp. 1. Grammaticality scores corresponding to this model are plotted in Fig. 4.



(a) Perception verbs (b) Attitude/action verbs

Figure 2: Distributions of the grammaticality scores for Exp. 2 with `gpt2-base-french`. Same notations as before.

5 Experiment 3

We finally test property (5) on 4 BERT-like LMs fine-tuned to perform natural language inference (see Table 9).

ID	Model	Lang.	Reference
12	camembert-base-xnli	fr	(Doy)
13	xlm-roberta-large-xnli-finetuned-mnli	multi	(Ozs)
14	mDeBERTa-v3-base-mnli-xnli	multi	(Laurer et al., 2022)
15	mDeBERTa-v3-base-xnli-multilingual-nli-2mil7	multi	(Laurer et al., 2022)

Table 9: Models used in Exp. 3

Given a negated matrix perception verb embedding a clause $\mathcal{C}$ either as an infinitive or as a (pseudo)relative, with or without cliticization of its subject (2×2 design, see (11)), we measure how likely LLMs are to infer the truth of $\mathcal{C}$ ("target inference", **TI**).

(11) Template for the stimuli of Exp. 3

$$\underset{\text{PRO}}{\left\{ \text{Il/Elle} \right\}} \quad \underset{\text{NEG}}{\text{ne}} \quad \underset{\text{(CL)}}{\left\{ \begin{array}{c} \text{le/la/l'} \\ \emptyset \end{array} \right\}} \underset{\text{V}}{\left\{ \text{voit/...} \right\}} \underset{\text{NEG–}}{\text{pas –}}$$

$$\underset{\text{(N)}}{\left\{ \begin{array}{c} \emptyset \\ \text{Marie/Jean} \end{array} \right\}} \quad \underset{\text{–}}{\text{–}} \quad \underset{\text{CP}}{\left\{ \begin{array}{c} \text{subject-gap relative} \\ \text{subject-gap infinitive} \end{array} \right\}}$$

(12) Example stimuli for Exp. 2

a. Il ne voit pas Marie qui danse.
He NEG sees NEG Marie that dances.
$\Longrightarrow$ Marie is dancing. TI ✓

b. Il ne la voit pas qui danse.
He NEG CL sees NEG that dances.
$\Longrightarrow$ She is dancing. TI ✓

c. Il ne voit pas Marie danser.
He NEG sees NEG Marie dancing.
$\not\Longrightarrow$ Marie is dancing. TI ✗

d. Il ne la voit pas danser.
He NEG CL sees NEG dancing.
$\not\Longrightarrow$ She is dancing. TI ✗

Sentence	`clitic?`	`emb_clause`
(12a)	n	relative
(12b)	y	relative
(12c)	n	infinitive
(12d)	y	infinitive

Table 10: Summary of the 2×2 design of Exp. 3

We expect the TI to be overall stronger when the embedded clause is a relative as opposed to an infinitive, whether of not the head noun is cliticized. The effect of cliticization in the case of a structure embedding a relative is a little bit less clear: in the absence of cliticization the clause is ambiguous between a PR and a RC, and it is reasonable to think that both parses encourage a TI. Assuming that the RC parse imposes a somewhat stronger TI than a PR parse, then we might expect sentences like (12b), which are unambiguously PRs due to cliticization, to lead to a slightly weaker TI than sentences like (12a) which allow a RC parse. In other words, we expect non-cliticized sentences embedding an RC to yield the strongest TI.

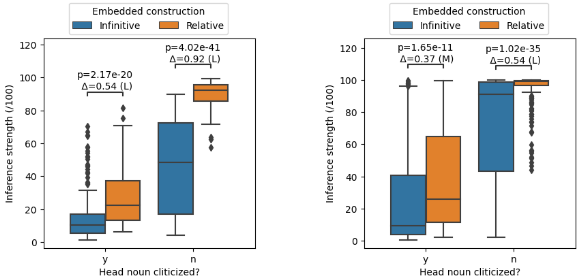
Linear mixed-effect modeling reveals that embedded relative constructions systematically lead to a stronger target inference as opposed to infinitives (cf. Table 11 col. 3), which is *consistent* with property (5), might be driven by the RC-parse only. Non-cliticized subjects also lead to a stronger target inference across the board (col. 4). This is made particularly clear in Figure 3. This pattern cannot be fully explained by the theory but makes sense if we consider that non-cliticized constructions are way more frequent in the data (so that LLMs may be more confident about the inferences related to such constructions, as opposed to cliticized ones). Finally, 2/4 models associate non-cliticized RC-embedding constructions to a stronger TI, which corresponds to the stipulation discussed in the previous paragraph. This all suggests that LLMs associate the target inference with the occurrence of RCs, but not really PRs: otherwise, *cliticized* relative constructions (unambiguously PRs) would have lead to stronger target inferences. Figure 3 in particular, shows that cliticized constructions featuring an embedded relative (unambiguously PRs), do not lead at all to a strong TI, suggesting the RC-parse (and not the PR-parse), is driving this inference.

ID	best AIC?	embedded clause (RC)	clitic	RC/clitic interaction
12	y	** (+)	** (-)	** (-)
13	y	** (+)	** (-)	** (+)
14	y	** (+)	** (-)	** (+)
15	y	** (+)	** (-)	** (-)

Table 11: Significance results of LME modeling for `target_inference_strength` $\sim$ `emb_clause` + `clitic` + `emb_clause` $*$ `clitic`.

6 Discussion and outlook

In this work, we investigated a structure (the pseudorelative) with two interesting distributional properties: (1) it can be ambiguous with a relative clause when the head noun is *not* cliticized; (2) the disambiguating (cliticized) structure is less frequent in corpora by several orders of magnitude. We think that the conjunction of these two properties makes learning the specific syntactic and semantic properties of PRs particularly challenging, even for models trained on large amount of data.



(a) CamemBERT (b) mDeBERTa

Figure 3: Distributions of the TI strength scores (/100) for Exp. 3 and models 12 and 15.

The experiments we run show that LLMs capture certain properties of PRs, pertaining to acceptable filler-gap dependencies, matrix verbs, and tense combinations. Interestingly, 3/4 multilingual models exposed to both French (a PR-language) and English (devoid of PRs) in Exp. 1 managed to contrast the two languages. Yet, the property that is perhaps the most specific to pseudorelatives, cliticization, does not seem to influence sentence probability scores in Exp. 2, and inference patterns in Exp. 3. This raises the question whether LLMs really get the specificity of the pseudorelative as a *syntactic construction* (Exp. 2) with a specific *semantics* (Exp. 3); or whether they simply recycle general processing heuristics applicable to other structures (e.g. standard RCs). Such heuristics may involve a preference for shorter dependencies (subject-gaps) across the board; or learning a statistical correlation between the use of perception verbs and the *agentive* structure of the perceived event.
Future work may involve investigating other languages allowing the pseudorelative, but also refining the current design by looking at the influence of the different perception verbs. We think this might be particularly relevant given the rather large frequency differences between these verbs in actual corpora (cf. Tables 1 and 2), and the potential imbalance between ambiguous vs. unambiguous PR-structures for each of those verbs.

References

camembert-base-xnli model card on huggingface. https://huggingface.co/BaptisteDoyen/camembert-base-xnli. Accessed: 2023-05-09.

gpt2-base-french model card on huggingface. https://huggingface.co/ClassCat/gpt2-base-french. Accessed: 2023-05-09.

xlm-roberta-large-xnli-finetuned-mnli model card on huggingface. https://huggingface.co/tuni/xlm-roberta-large-xnli-finetuned-mnli. Accessed: 2023-05-09.

Julien Abadji, Pedro Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2022. Towards a Cleaner Document-Oriented Multilingual Crawled Corpus. *arXiv e-prints*, page arXiv:2201.06642.

Julien Abadji, Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2021. Ungoliant: An optimized pipeline for the generation of a very large-scale multilingual web corpus. Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-9) 2021. Limerick, 12 July 2021 (Online-Event), pages 1 – 9, Mannheim. Leibniz-Institut für Deutsche Sprache.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Isaac Caswell, Julia Kreutzer, 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 Javier Ortiz Suárez, Iroro Orife, Kelechi Ogueji, Rubungo Andre Niyongabo, 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. 2021. Quality at a Glance: An Audit of Web-Crawled Multilingual Datasets. *arXiv e-prints*, page arXiv:2103.12028.

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. *CoRR*, abs/1911.02116.

Aniello De Santo and So Young Lee. 2022. Evaluating structural economy claims in relative clause attachment. In *Proceedings of the Society for Computation in Linguistics 2022*, pages 65–75, 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. *CoRR*, abs/1810.04805.

M. T. Guasti. 1988. La pseudorelative et les phénomènes d'accord. *Rivista di Grammatica Generativa*, 13:35–80.

John Hale. 2001. A probabilistic earley parser as a psycholinguistic model. In *Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies 2001 - NAACL '01*. Association for Computational Linguistics.

Richard S. Kayne. 1975. *French syntax. The transformational cycle.* MIT Press, Cambridge (MA).

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Moritz Laurer, Wouter van Atteveldt, Andreu Salleras Casas, and Kasper Welbers. 2022. Less Annotating, More Classifying – Addressing the Data Scarcity Issue of Supervised Machine Learning with Deep Transfer Learning and BERT - NLI. *Preprint*. Publisher: Open Science Framework.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2020. Flaubert: Unsupervised language model pre-training for french. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France. European Language Resources Association.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. Camembert: a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2022. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006, Seattle, United States. Association for Computational Linguistics.

Kanishka Misra. 2022. minicons: Enabling flexible behavioral and representational analyses of transformer language models. *arXiv preprint arXiv:2203.13112*.

Keir Moulton and Nino Grillo. 2015. Pseudo relatives: Big and direct. In *Proceedings of 45 North Eastern Linguistic Society*, pages 193–202. MIT.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoit Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Pedro Javier Ortiz Suárez, Benoit Sagot, and Laurent Romary. 2019. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pages 9 – 16, Mannheim. Leibniz-Institut f"ur Deutsche Sprache.

Céline Pozniak, Barbara Hemforth, Yair Haendler, Andrea Santi, and Nino Grillo. 2019. Seeing events vs. entities: The processing advantage of pseudo relatives over relative clauses. *Journal of Memory and Language*, 107:128–151.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Andrew Radford. 1975. Pseudo-relatives and the unity of subject raising. *Archivum Linguisticum*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *CoRR*, abs/1806.03822.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250.

C Schwarze. 1974. 'les constructions du type "je le vois qui arrive". In *Actes du Colloque Franco-Allemand de Grammaire Transformationnelle*, pages 18–30, Tübingen. Niemeyer.

Skipper Seabold and Josef Perktold. 2010. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *CoRR*, abs/1905.00537.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *CoRR*, abs/1804.07461.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Preparing a corpus of spoken Xhosa

Eva-Marie Bloom Ström[1], Onelisa Slater[2], Aron Zahran[1], Aleksandrs Berdicevskis[3] and Anne Schumacher[3]

[1]Department of Philosophy, Linguistics and Theory of Science, University of Gothenburg
[2] Department of Linguistics and Applied Language Studies, Rhodes University
[3] Språkbanken Text, Department of Swedish, Multilingualism, Language Technology, Gothenburg University

eva-marie.strom@gu.se, onelisaslater@outlook.com, aron.zahran@hotmail.com, aleksandrs.berdicevskis@gu.se, anne.schumacher@svenska.gu.se

Abstract

The aim of this paper is to describe ongoing work on an annotated corpus of spoken Xhosa. The data consists of natural spoken language and includes regional and social variation. We discuss encountered challenges with preparing such data from a lower-resourced language for corpus use. We describe the annotation, the search interface and the pilot experiments on automatic glossing of this highly agglutinative language.

1 Introduction

Xhosa, or isiXhosa, is a Bantu language of the Nguni sub-group, spoken in South Africa. Approximately 16 percent of South Africa's population speak the language as their first language, and it is one of 11 official languages of the country (Statistics South Africa 2012). Xhosa is to a large extent mutually intelligible with the other Nguni languages Ndebele, Swati and especially Zulu. Although a relatively large language, it can be considered a lower-resourced language in several respects including in terms of its digital resources. There exist unannotated text collections made available through the South African Centre for Digital Language Resources (SADiLaR), and since recently also an annotated parallel corpus for the four Nguni languages (Gaustad and Puttkammer 2022). The corpus consists of ca. 50 000 tokens of government texts for each language (translated from English) (Gaustad and Puttkammer 2022). Annotated spoken language corpora are lacking altogether. There exists a small collection of audio resources (available through SADiLaR) such as orthographically transcribed audio recordings (6 hours) for the development of text-to-speech (Louw and Schlünz 2018). None of these resources contain natural conversation data. The aim of the current project is to fill this gap by creating an annotated corpus of spoken Xhosa. One important reason for this is that many speakers who were recorded, especially those belonging to minority communities in the area, requested that their contributions of data be well preserved and disseminated.

Consequently, a collaboration was initiated with the aim to make the data available and searchable. Besides providing the digital infrastructure, we aim to explore the possibilities of reducing the manual workload by using automated annotation tools.

2 Fieldwork and content of the data

The recordings included in the corpus all stem from fieldwork by the first author. These recordings have been made in different parts of the Eastern Cape, the province in South Africa where a majority of the population speak Xhosa. Not all speakers identify as Xhosa, however, since the identification as Xhosa implies a certain ancestral line. They identify as belonging to other communities with their own languages. In present-day South Africa, however, differences between these varieties are small as evidenced from the collected data (Bloom Ström 2018). Our material is therefore not necessarily in accordance with standard Xhosa norms. This gives a unique opportunity to study the language in all its facets, as the language is actually used in the communities. The recordings vary in spontaneity. The collection of texts includes dialogues with several speakers. Some of these dialogues are about a certain topic and others are

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 62–67
September 11–12, 2023.

completely free. There are monologues in which one speaker explains a certain procedure (e.g., cooking), or tells a traditional story, mostly including an audience. A minority of recordings are more controlled and based on stimuli, i.e., the speaker explains the content of a series of pictures or a film.

This is still a small corpus, with the aim of expanding when the infrastructure is in place. At present, there are approximately 10 hours of transcribed recordings. This is estimated to sum up to ca. 40 000 tokens. Metadata for each recording is noted, including the date, location, speaker information, topic of discussion, length of recording, and audio quality.

3 Premises for data preparation

The overall guideline in the process of making the data available has been maximal searchability for linguistic researchers.

3.1 Transcription Process

The time-consuming transcription process by language students at Rhodes University ensures that recorded audio is represented as accurately as possible in written form. Although standard orthography has been used for transcriptions, we take a descriptive approach to language. This means that we do not adjust the transcribed speech to prescriptive norms.

The idea is that this approach will provide potential corpora users with a rich set of data in which one can investigate things like phonological and/or morphosyntactic variation, but also potential developments and grammaticalization processes based on systematic distribution of different forms encoding the same function. A good illustration of this concerns future tense marking, see Example (1) (glossing follows the Leipzig glossing rules (Comrie et al. 2008/2015); abbreviations are listed at the end of this paper). A construction that originally involved the auxiliary verb *-za* 'come' followed by a verb in the infinitive, has then evolved into a verb form that to different degrees retains the infinitival marker *uku-/ku-*.

1. a) ba-**za** **uku**-fika
   SM.2-come INF-arrive
   'They will arrive'

   b) *si-**zaku**-ya* *kwa-malume!*
   SM.1PL-come.INF-go LOC-1a.uncle
   'We are going to (our) uncle!'

While in the utterance (1a) we can assume, based on phonological criteria, that the verb *-za* and the infinitival marker *-uku* follow each other as segmentable morphemes, in (1b) they are fused into a non-segmentable future marker. Further evidence for this fusion or grammaticalization is that this future marker, originating in a verb meaning 'to come', can in (1b) unproblematically be used with a verb 'to go' due to semantic bleaching of the original meaning of *-za*.

Hence, due to the variation in our data in the realization of this marker, e.g., *zu-, zaku-, zoku-, za-, zo-, zau-,* the grammaticalization process can be investigated. This variation is likely to be higher in our spoken data, than if the corpus was based on standardized written Xhosa.

3.2 Annotation

The morphemic annotation, or glossing, has proven to be a challenge since many areas of Xhosa grammar remain un(der)described. Deciding on a suitable translation for a certain morpheme has more often than not implied thorough investigation of available publications on the language, in combination with our own analysis together with mother tongue speaker and team member Onelisa Slater. There is no modern and comprehensive reference grammar of the language in which one can search for the right abbreviation. All decisions have been made with consideration to the Leipzig glossing rules (Comrie et al. 2008/2015), while also adhering to conventions used by researchers in Bantu linguistics. Ensuring searchability includes, for example, finding a balance between making the glossing general enough to include comparable forms, but also specific enough for the user to be able to unambiguously find what they are looking for. One example is the so-called augment, a vowel that in certain environments occurs before the noun class prefix or the nominal root. While it can certainly be interesting to consider all occurrences of the augment, the researcher might also be interested in only looking at the occurrences of the augment in more restricted settings, say in one specific noun class at a time. Since the augment itself is not noun class specific, search features can be combined to include only those augments that are followed by a nominal prefix or root of a certain noun class.

Another challenge with the glossing stems from the fact that surface forms of (especially spoken) Xhosa do not always show all the information contained in the underlying form, for example because of vowel elision. For this reason, we make use of underlying forms in our glossing, while also showing the surface form in transcription. Again, the augment provides an interesting case in point. In example (2a), the vowel of the comitative marker *na-* coalesces with the augment vowel *i*, forming *e*. In example (2b), the augment vowel of noun class 6 is *a*, i.e. the same as the vowel of the comitative. In (2b), it is therefore not transparent in the surface form that the augment occurs, although it would definitely be in the interest of the researcher to find these constructions as well, when looking for environments with the augment.

2. a) badibana *nendoda*
   ba-dib-an-a na-i-ndoda
   SM.PST.2-meet-RECP-FV COM-AUG-9.man
   'they met with a man'

   b) *namakhwenkwe*
   na-a-ma-khwenkwe
   COM-AUG-NCP.6-6.boy
   'with the boys'

While this is a very effective way of making forms more transparent to the user, and making underlying morphemes searchable in the corpus, it also requires further analysis and decision making on the extent to which these underlying forms can be safely assumed.

Moreover, considerations are made on how the glossing conventions can be combined with part-of-speech (POS) tags when searching in the corpus, as these combinations can serve to make searches more inclusive or exclusive depending on the aim of the user. POS tags add information that is not encoded in the glossing, which could help the potential corpus user to identify the functions of different constructions in Xhosa. In cases where tokens are homonymous, POS-tagging can help disambiguate. Example (3), for instance, demonstrates that the token *ukuhamba* from the lexical root *hamb-* 'walk', can be labelled either as verb or a noun based on its syntactic properties. In (3a) *ukuhamba* is a verbal noun/gerund, tagged as a noun in the corpus, while in (3b) it is a verbal infinitive following the inflected first verb 'want' and tagged as a verb:

3. a) *u-ku-hamba* *kw-am*
   AUG-NCP.15-walk 15-POSS.1
   'my walking'

   b) *ndi-fun-e* *uku-hamba*
   SM.1SG-want-REC.CJ INF-walk
   'I wanted to walk'

One of the main challenges in this regard has been the universality of established part of speech categories, and to what extent tags like the ones used by Universal Dependencies (de Marneffe et al. 2021) are applicable to Xhosa. A relevant example concerns non-verbal predication in Xhosa, in which a copula is prefixed to a noun as in example (4). The copula *ngu-* is verb-like in that it takes some inflectional morphology, although there is not enough diachronic nor synchronic evidence for it to be considered a verb. For example, it does not possess other verbal properties like taking derivational morphology or having an infinitival form. Tagging the whole construction as either a copula or a noun would however not make it justice, but rather, we identify the need of a specialized part-of-speech category called "nominal copula"; NCOP in this case (while the morpheme abbreviation remains COP):

4. Ndandi-ngu-m-ntu
   SM.PST.IPFV.1SG-COP.1-NCP.1-1.person
   'I am a person'

4 Pilot experiments on automatic annotation

Automatic annotation of spoken Xhosa texts faces several challenges: first, the small amount of data available, second, frequent variation and usage of non-standard forms. Third, the annotation guidelines are being finalized as the manual annotation progresses, which means that the tag sets have not been finalized yet. Despite that, we make a preliminary attempt to estimate whether parts of the pipeline can be automated.

As mentioned above, a corpus of written Xhosa (Gaustad and Puttkammer 2022) has recently been released, and an annotation tool used to create it have also been made available by SADiLaR (du Toit and Puttkammer 2021). SADiLaR, however, uses different glossing

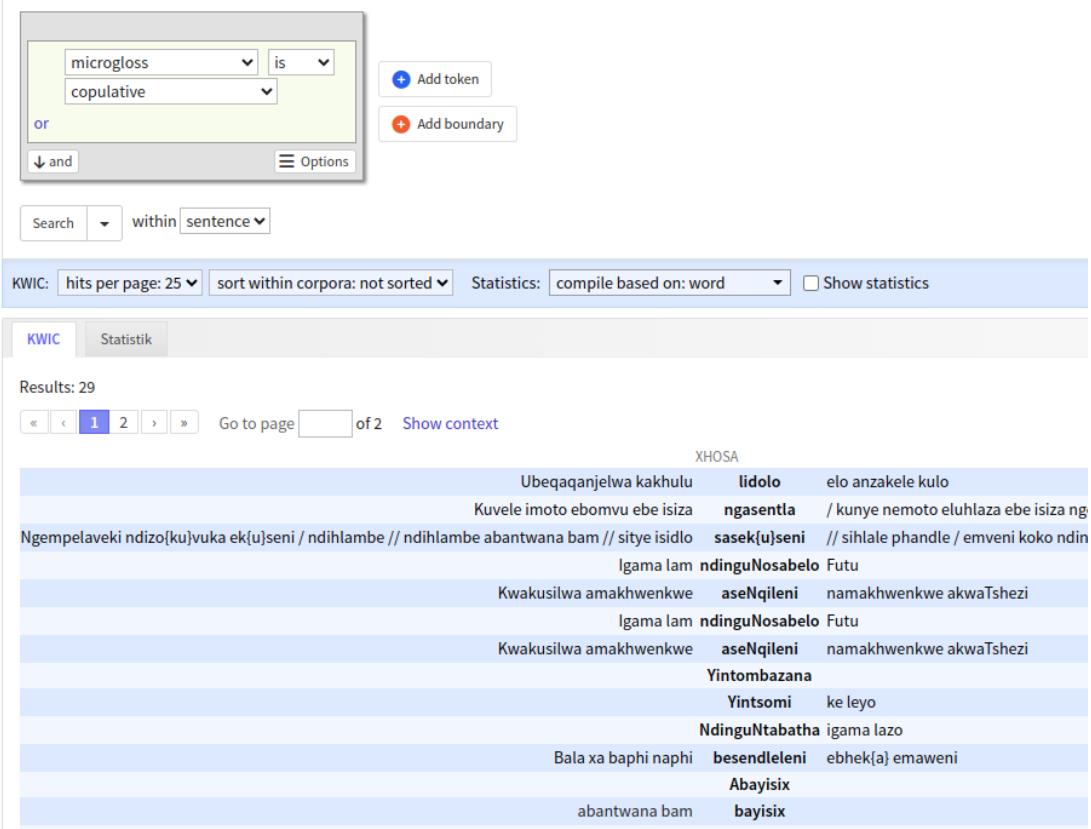


Figure 1: Search for the copulative

principles. The POS tag set, on the other hand, was judged to be compatible with the purposes of the current project. Du Toit and Puttkammer (2021) report the accuracy of their POS tagger, based on the Marmot tagger (Mueller et al. 2013) and trained on the parliamentary texts, to reach 96% in the same domain. On our data, the accuracy is 74%. The drop in accuracy is unsurprising, given the high number of out-of-vocabulary items and the systematic differences in the usage of grammatical forms.

Since the SADiLaR corpus cannot be used to train a morphemic (glossing) tagger, we ran a pilot experiment, training Marmot on our own data. Despite a very small training set of 1122 morphemes, Marmot achieves 67% on the test set (267 morphemes). As is common in such tasks (Barriga Martínez et al. 2021), we did not attempt glossing stems, using the LEX tag for all stems instead.

On the grammatical morphemes only, the accuracy is 51%, with some of the ambiguous morphemes being correctly tagged.

Pre-annotating the texts automatically and manually post-correcting them is likely to be more efficient than manually annotating them from scratch. As the amount of manually annotated data increases, the performance of the tagger will hopefully improve. It remains to be seen whether, given the small training set, “fast learners” like Marmot can be beaten by large languages models (e.g. Eiselen 2023), fine-tuned on the same data.

We have not yet attempted automatically segmenting words into morphemes.

5 The search interface

The corpus is hosted by Språkbanken Text (SBX) and available[1] through the corpus search tool Korp (Borin et al. 2012). Korp can be used to perform advanced corpus search queries where

[1] https://spraakbanken.gu.se/korp/?mode=xhosa

transcriptions along with their annotations (segmentation, glosses, POS, lexical meanings etc.) can be used as search parameters. The parameters can be combined in various ways in order to refine the search.

Note that the parameters apply to different levels of analysis; some are on sentence level (e.g., idiomatic translation of the whole sentence), some are on token level (e.g., POS, lexical meaning), some are on sub-word (morpheme) level (gloss).

For querying purposes, we distinguish between “microglosses” and “macroglosses”. A microgloss is any single gloss, the smallest possible unit of glossing, e.g., PST: ‘past tense’. Macrogloss is any gloss of a non-segmentable morph. It may contain one microgloss (e.g., RECP for *an* in example 2a) or several microglosses if the morph expresses several grammatical meanings at once, e.g., SM.PST.2 (the gloss for *ba* in example 2a) or SM.PST.IPFV.1SG (the gloss for *ndandi* in example 4). Depending on the users’ needs, they may either search for a micro- or macrogloss. The search for microgloss PST, for instance, would return both example (2a) and (4), but it is also possible to search specifically for the macrogloss SM.PST.2.

As an example, Figure 1 shows a search for all copulatives in the corpus (COP).

For this particular corpus a special button was added to the interface which allows the user to copy a traditional four-row representation of glossed examples in linguistics (surface form, underlying form, glossing and translation, cf. example 2b). This was done to facilitate using examples from search results in publications or for teaching purposes.

The corpus will be publicly available, both in Korp and as a downloadable data set.

In the future we will also incorporate the original audio recordings into Korp, and, ideally, synchronize them with the transcriptions (cf. the implementation in the IVIP corpus [2]).

Limitations

The limitations of this project first and foremost concern the amount of data. As automatic annotation starts to improve, the idea is to keep adding transcribed texts to the corpus and this is expected to improve accuracy. Further tests of different kinds of automatic annotation are required.

Ethics Statement

The project includes data provided by speakers of the relevant language and varieties. The conversations concern everyday issues of the people, but also traditional stories and more controlled speech based on stimuli. Recordings which nevertheless ended up containing possibly sensitive information have been removed. All speakers have given informed consent for the use of the recorded data for research and publication purposes.

Acknowledgments

The authors are immensely thankful to all speakers who have contributed data; transcribers; as well as to those who have assisted with contacts and have helped during recording sessions.

In the meticulous process of researching the grammar of Xhosa, we are thankful to Stefan Saviç, Thera Crane, Jochen Zeller and Fahima Ayub Khan for generously sharing their expertise.

We thank the South African Centre for Digital Language Resources (SADiLaR) for their time and assistance at different stages of this project.

The Swedish Research Council is gratefully acknowledged in making this data collection and the spoken language corpus possible in funding the following projects: Morphosyntactic variation in the dialects of Xhosa (VR2014-00244); The role of the verb phrase and word order in the expression of definiteness in Bantu languages (VR2017-01811); How do words get in order? The role of speaker-hearer interaction in languages of southern Africa (VR2021-03125). For the development of the corpus, this work has been supported by Nationella språkbanken – jointly funded by its 10 partner institutions and the Swedish Research Council (VR 2017-00626).

Abbreviations

AUG augment, a nominal prefix combined with the noun class prefix
CJ conjoint; one of two morphological forms in certain tenses
COM comitative

[2] https://spraakbanken.gu.se/korp/#?corpus=ivip-demo

COP	copulative
FV	final vowel, indicative mood
INF	infinitive prefix
IPFV	imperfective
LOC	locative
NCP	noun class prefix
POSS	possessive
PST	past
REC	recent past
RECP	reciprocal
SM	subject marker

Numbers not followed by SG or PL identify noun class.

References

Barriga Martínez, Diego, Victor Mijangos and Ximena Gutierrez-Vasques. 2021. Automatic Interlinear Glossing for Otomi language. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 34–43, Online. Association for Computational Linguistics.

Bloom Ström, Eva-Marie. 2018. Linguistic and sociolinguistic aspects of variation in the Eastern Cape: complexities of Xhosa language use. *Studia Orientalia Electronica* 6:90-120.

Borin, Lars, Markus Forsberg, Johan Roxendal. 2012. Korp – the corpus infrastructure of Språkbanken. In *Proceedings of LREC 2012*. Istanbul: ELRA, volume Accepted, pages 474–478.

Comrie, Bernard, Martin Haspelmath and Balthasar Bickel. 2008, updated 2015. The Leipzig Glossing Rules: Conventions for interlinear morpheme-by-morpheme glosses, edited by M. P. I. f. E. Anthropology. Leipzig.

de Marneffe, Marie-Catherine, Christopher D. Manning, Joakim Nivre and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics* 2021; 47 (2): 255–308.

du Toit, Jakobus S. and Martin J. Puttkammer. 2021. Developing core technologies for resource-scarce Nguni languages. *Information* 2021 *12*, 520.

Eiselen, Roald. 2023. NCHLT isiXhosa RoBERTa language model. North-West University; Centre for Text Technology (CTexT).

Gaustad, Tanja and Martin J. Puttkammer. 2022. Linguistically annotated dataset for four official South African languages with a conjunctive orthography: IsiNdebele, isiXhosa, isiZulu, and Siswati. *Data in brief* 41.

Louw, Aby and Georg Schlünz. 2018. Lwazi III isiXhosa TTS Corpus. Meraka Institute, CSIR.

Mueller, Thomas, Helmut Schmid and Hinrich Schütze. 2013. Efficient Higher-Order CRFs for Morphological Tagging. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 322–332, Seattle, Washington, USA. Association for Computational Linguistics.

Statistics South Africa. 2012. *Census 2011, census in brief.* Retrieved from www.statssa.gov.za.

Machine Translation of Folktales: small-data-driven and LLM-based approaches

Olena Burda-Lassen, Ph.D.
Independent Research Scientist, Colorado, United States
oburdalassen@gmail.com

Abstract

Can Large Language Models translate texts with rich cultural elements? How "cultured" are they? This paper provides an overview of an experiment in Machine Translation of Ukrainian folktales using Large Language Models (Open AI), Google Cloud Translation API, and Opus MT. After benchmarking their performance, we have fine-tuned an Opus MT model on a domain-specific small dataset specially created to translate folktales from Ukrainian to English. We have also tested various prompt engineering techniques on the new Open AI models to generate translations of our test dataset (folktale 'The Mitten') and have observed promising results. This research explores the importance of both small data and Large Language Models in Machine Learning, specifically in Machine Translation of literary texts, on the example of Ukrainian folktales.

1 Introduction

"ChatGPT has already become a good translator" (Jiao et al., 2023) is an increasingly popular statement. We see an exponential increase in using Open AI models for various Machine Learning tasks and wanted to further explore this new tendency.

In addition to human translation, machine translation has undeniable potential in connecting people and cultures. Therefore, improving accuracy and accessibility to high-quality machine translation tools is very important.

We chose Ukrainian folktales for this experiment due to their unique nature and rich linguistical ecosystem. Folktales are usually passed on from one generation to another, going back hundreds and sometimes thousands of years, creating immense depth of knowledge and layers of cultural relevance.

The Ukrainian language has an extensive collection of myths, legends, proverbs, songs, and folktales. Even though these texts have literary translations available, many of them are rather transcreations, meaning that stories are retold and adapted to the target language and culture.

This experiment uses a recently created corpus of domain-specific curated parallel training data: Ukrainian-To-English Folktale Corpus (Burda-Lassen, 2022).

We wanted to use this curated corpus for fine-tuning machine translation models and see the impact on the accuracy and quality of translation.

2 Machine Translation Process

2.1 Overview of resources and machine translation models

For the creation of the Ukrainian-To-English Folktale Corpus, we used familiar Ukrainian folktales that were available in English: folktales from various websites for children's literature, blogs about Ukrainian traditions, bilingual children's books, as well as English translations from the Gutenberg Project[1].

Training a reliable machine translation system requires a large number of parallel sentences in two languages, which is often widely unavailable in low-resource language pairs (Sánchez-Cartagena

[1] https://www.gutenberg.org/cache/epub/29672/pg29672.txt

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 68–71
September 11–12, 2023.

et al., 2021). Even though Ukrainian is not considered a low-resource language anymore, the availability of the Ukrainian-To-English Folktale Corpus was very helpful in our experiment.

2.2 Applied Methods

Since the focus of our research was comparing the performance of 3 machine translation approaches and models for the translation of Ukrainian folktales, we have selected specific cultural terms that would be more domain-specific and, therefore, more challenging to translate.

While most common phrases are already being translated accurately by available machine translation engines, rare or cultural terms are often mistranslated or generalized. Adding an extra layer of culturally significant information can significantly improve the outcome of the translation process.

We have tested the translation at the word and text levels. We have chosen a subset of the words that have a high level of cultural sensitivity and are challenging to translate (Table 1).

For text-level translation, we have chosen as our test dataset the Ukrainian folktale 'The Mitten.' We have translated this folktale into English, using carefully selected human translation techniques, to preserve culturally specific elements, their meaning, and literary style. We have then translated this text using Google Cloud Translation API, the pre-trained model 'Helsinki-NLP/opus-mt-uk-en' and 2 Open AI models ('text-davinci-002' and the more recent 'gpt-3.5-turbo-16k'). We have also fine-tuned 'Helsinki-NLP/opus-mt-uk-en' on the Ukrainian-To-English Folktale Corpus and tested the accuracy of the translation by the fine-tuned version of the model. We have used sacreBLEU and BERTScore as evaluation metrics.

Ukrainian	English: human translation	Open AI: text-davinci-002	Google Cloud Translation API	Opus MT	Open AI: gpt-3.5-turbo-16k
мишка-шкряботушка	Scratchy-Mouse	Mouse-squeak	Mouse-scratcher	Roaster mouse	Bear-scratchy
жабка-скрекотушка	Croaky-Frog	toad-croaker	frog-scratch	snot frog	frog-squeaky
ведмідь-набрідь	Bear, the Wanderer	badger-beard	bear-bear	bear-brick	bear-wanderer
кабан-іклан	Boar-With-Tusks	wild boar-tusks	wild boar	boar-aklan	boar-tusk
коза-дереза	Bully Goat	goat-bristles	goat-skunk	goat-tree	goat-stick
солом'яний бичок	a little straw bull	straw-haired hog	straw bull	Somus-Treaby	Straw bull
бичок-третячок	Three-year-old bull calf	third-class hog	bull-tretyachok	monkey	bull-third
Мавка	Mavka, the forest spirit	fly-agaric	magpie	monkey	Mavka wood nymph
мед-вино	honey-wine	honey-wine	honey-wine	honey wine	honey-wine

Table 1: Translation examples of selected culture-specific terms.

2.3 Key Findings

After reviewing machine translation predictions, we can identify a few specific translation techniques and tendencies: calque (loan translation), generalization, and transcription.

As we can see in Table 1, many examples were mistranslated (especially the word to describe one of the female spirits in Ukrainian mythology *Mavka, the forest spirit*). If a term consisted of commonly known words, it was translated more precisely. Therefore, all tested models heavily rely on general language corpora and do not predict the values of specialized terminology, which formalizes the need for fine-tuning these models on small datasets that are very specific and rich in such valuable domain information.

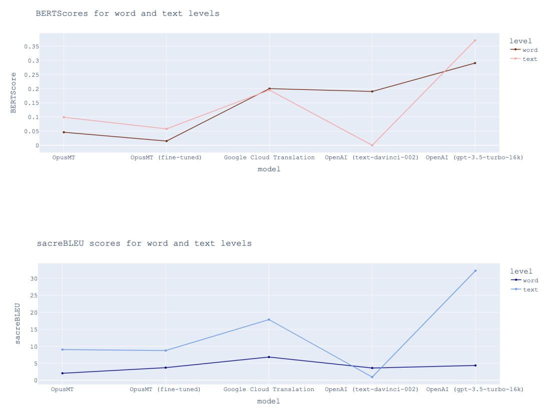


Table 2: BERTScore and sacreBLEU scores for models at text and word levels.

Word level measurements refer to the accuracy of translation of the specific subset of cultural terms, while text level accuracy evaluates the translation of the folktale 'The Mitten.'

We have observed an increase in sacreBLEU score on the word level after fine-tuning the Opus MT model (from *2.05* to *3.70*), and not on the text level. The highest accuracy in translating 'The Mitten' was observed by running predictions using the Open AI model 'gpt-3.5-turbo-16k'. We noticed that it outperformed all other models on the text level, and not on the word level (where Google Cloud Translation API was in the lead).

However, BERTScore showed the highest accuracy values for word and text levels for Open AI 'gpt-3.5-turbo-16k', with Google Cloud Translation API consistently showing second-best accuracy scores.

Additionally, we have noticed a rather significant difference between the two tested Open

AI models. While no notable accuracy improvement was observed from using ‘text-davinci-002’ at the word level translation, the translation of ‘The Mitten’ was unsuccessful: the story was simply summarized into eight sentences with most of the plot missing.

Open AI model ‘gpt-3.5-turbo-16k’, on the other hand, performed really well. We have applied several prompt engineering techniques, and the model was able to translate text according to the requirements of applying different styles. For example, when prompted with *‘This text is for a children's book and needs to be fun and interactive.\ Create interesting compound words.’*, it created new compound words, very similar to the ones from the human translation. While prompted with ‘*This text is informational for English-speakers to learn more about Ukrainian culture*’, the model translated all terms as a combination of two very appropriate human translation techniques: transliteration with an explanation.

Here are a few examples:

"Mishka-shkryabotushka" - A term used to describe a small bear that scratches itself.
"Zhabka-skrekotushka" - A term used to describe a small frog that makes a screeching sound.
"Vedmid-nabrid" - A term used to describe a bear that is annoying or bothersome.
"Kaban-iklan" - A term used to describe a boar with long tusks.
"Koza-dereza" - A term used to describe a goat with sharp horns.
"Solom'yanyi bychok" - A term used to describe a scarecrow made of straw.
"Bychok-tretyachok" - A term used to describe a small bull or young bull.
"Mavka" - A term used to describe a mythical creature from Ukrainian folklore, often depicted as a forest nymph or spirit.
"Med-vyno" - A term used to describe mead, an alcoholic beverage made from fermented honey.

Even though there is an error in translating the term ‘*Mishka-shkryabotushka’* (it is a small mouse, not a bear), with the correct transliteration being ‘*Myshka-shkryabotushka’,* this definitely was an interesting machine translation output, which calls for further study and research.

While small datasets with domain-specific information can help train the traditional neural machine translation models and increase accuracy, especially if examples are carefully curated and hand-picked, Large Language Models have the potential to increase translation accuracy and create style-specific translations.

3 Conclusion

More research is necessary to increase the size of the Ukrainian-To-English Folktale Corpus to include a broader range of cultural terms, which will help further explore the preferable size of small data to make a more noticeable impact on accuracy score.

Since we have noticed an increase in translation accuracy at the word level after fine-tuning an Opus MT model, it would be valuable to explore the depth and volume of cultural terms needed to increase the accuracy score even further.

Another area of research could be prompt engineering and fine-tuning LLMs, while exploring their added benefit of creating machine translation tailored to specific literary styles.

Contrary to human translation of folklore, machine translation techniques must be more literal and descriptive. Therefore, a significant difference exists between human and machine translation techniques for folktales. That’s where using a more informational translation style could be very valuable.

We believe that this type of research would be important for other language pairs as well. The domain of literary translation, specifically the translation of folklore and other culturally specific texts, is a vibrant environment full of fascinating challenges and great potential.

References

Eleftherios Avramidis, Marta R. Costa-jussà, Christian Federmann, Josef van Genabith, Maite Melero, and Pavel Pecina. 2012. A Richly Annotated, Multilingual Parallel Corpus for Hybrid Machine Translation. In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), pages 2189–2193, Istanbul, Turkey. European Language Resources Association (ELRA).

Olena Burda-Lassen. 2022. Ukrainian-To-English Folktale Corpus: Parallel Corpus Creation and Augmentation for Machine Translation in Low-Resource Languages. In Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Workshop 2: Corpus Generation and Corpus Augmentation for Machine Translation), pages 28–31, None. Association for Machine Translation in the Americas.

Dmitriy Genzel, Jakob Uszkoreit, and Franz Och. 2010. “Poetic” Statistical Machine Translation: Rhyme and Meter. In Proceedings of the 2010 Conference on Empirical Methods in Natural

Language Processing, pages 158–166, Cambridge, MA. Association for Computational Linguistics.

Natalia Grabar, Kanishcheva Olga, Hamon Thierry. Multilingual aligned corpus with Ukrainian as the target language. SLAVICORP, Sep 2018, Prague, Czech Republic. ffhalshs-01968343f

Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. 2023. Is chatgpt a good translator? a preliminary study. arXiv preprint arXiv:2301.08745.

Víctor M. Sánchez-Cartagena, Miquel Esplà-Gomis, Juan Antonio Pérez-Ortiz, and Felipe Sánchez-Martínez. 2021. Rethinking Data Augmentation for Low-Resource Neural Machine Translation: A Multi-Task Learning Approach. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 8502–8516, Online and Punta Cana, Dominican Republic. 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.

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.

Yao, B., Jiang, M., Yang, D., & Hu, J. (2023). Empowering LLM-based Machine Translation with Cultural Awareness. arXiv preprint arXiv:2305.14328.

Yıldız, Eray & Tantuğ, Ahmet & Diri, Banu. (2014). The Effect of Parallel Corpus Quality vs Size in English - Toturkish SMT. Computer Science & Information Technology. 4. 21-30. 10.5121/csit.2014.4710.

Example-Based Machine Translation with a Multi-Sentence Construction Transformer Architecture

Haozhe Xiao **Yifei Zhou** **Yves Lepage**
IPS, Waseda University
2-7 Hibikino, Wakamatsu-ku, Kitakyushu-shi, 808-0135 Fukuoka-ken, Japan
xiaohaozhe@moegi.waseda.jp, yifei.zhou@ruri.waseda.jp, yves.lepage@waseda.jp

Abstract

Neural Machine Translation (NMT) has now attained state-of-art performance on large-scale data. However, it does not achieve the best translation results on small data sets. Example-Based Machine Translation (EBMT) is an approach to machine translation in which existing examples in a database are retrieved and modified to generate new translations. To combine EBMT with NMT, an architecture based on the Transformer model is proposed. We conduct two experiments respectively using limited amounts of data, one on an English-French bilingual dataset and the other one on a multilingual dataset with six languages (English, French, German, Chinese, Japanese and Russian). On the bilingual task, our method achieves an accuracy of 96.5 and a BLEU score of 98.8. On the multilingual task, it also outperforms OpenNMT in terms of BLEU scores.

1 Introduction

An analogy is a relationship between four objects, A is to B as C is to D. Studies on analogies have investigated their utility in different applications, like machine translation. Solving analogies between sentences involves the task of generating an unknown D that satisfies an analogical equation $A : B :: C : D$, where A, B, and C are given. Here is an example of a sentence analogy:

$$\textit{he 's coming .} : \textit{i am coming .} :: \begin{array}{l}\textit{he 's eating}\\ \textit{an apple .}\end{array} : x$$

$$\Rightarrow \quad x = \begin{array}{l}\textit{i am eating}\\ \textit{an apple .}\end{array}$$

EBMT extracts knowledge from a corpus in two languages to perform translation. Concretely, the process of EBMT by analogy involves extracting analogical relationships in the source language to find the corresponding sentences in the target language and solve a sentence analogy.

Formula (1) defines the notation of analogies between sentences in two languages. As instantiated in Formula (2), the translation result for "*i am eating an apple .*" is "*je manger une pomme .*", which can be obtained through the reasoning process.

$$\begin{array}{ccccccc} A & : & B & :: & C & : & D \\ \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow \\ A' & : & B' & :: & C' & : & D' \end{array} \tag{1}$$

$$\begin{array}{ccccccc} \textit{he 's coming .} & : & \textit{i am coming .} & :: & \textit{he 's eating an apple .} & : & \textit{i am eating an apple .} \\ \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow \\ \textit{il est en train d' arriver .} & : & \textit{j' arrive .} & :: & \textit{il est en train de manger une pomme .} & : & ?? \end{array} \tag{2}$$

EBMT by analogy is a translation method that involves generating a target translation by using multiple example sentences for reference and reasoning. However, the vanilla Transformer (Vaswani et al., 2017) model can only handle one input at a time. To address this limitation, we propose a multi-sentence construction Transformer architecture designed specifically for EBMT by analogy.

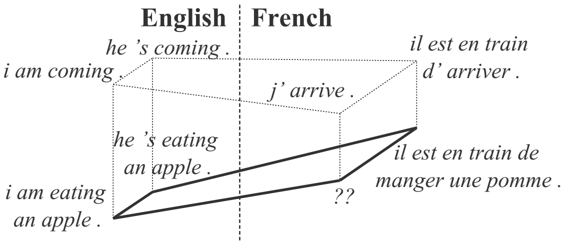
2 Previous work and proposal

To perform translation, Nagao (1984) proposed an approach to EBMT that considers a bilingual analogy across two languages. Translations are made by transferring symbolic knowledge from the source language to the target language. In Figure 1(a), the translation of "*i am eating an apple .*" is achieved by solving a bilingual analogy:

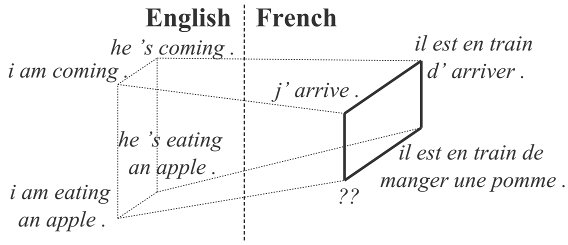
$$\begin{array}{l}\textit{i am}\\ \textit{coming .}\end{array} : \textit{j' arrive .} :: \begin{array}{l}\textit{i am eating}\\ \textit{an apple .}\end{array} : ??$$

Figure 1(b) outlines the indirect approach to EBMT. As previously shown in Formula (2), previous research considered two monolingual analogies in two different languages that correspond to

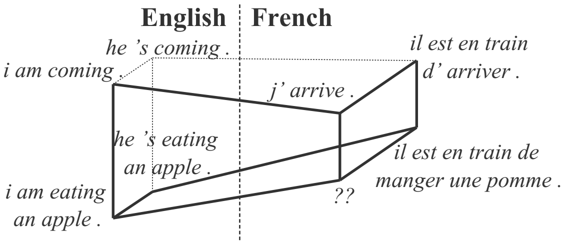
Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 72–80
September 11–12, 2023.



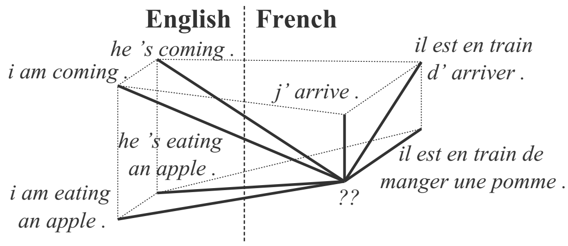
(a) direct approach (Nagao, 1984)



(b) indirect approach (Lepage and Denoual, 2005)



(c) combination of direct and indirect approaches (Taillandier et al., 2020)



(d) our proposed method

Figure 1: Different approaches to EBMT by analogy (adapted from (Taillandier et al., 2020)). In each sub-figure, the left half shows the embedding space for English sentences while the right half shows the embedding space for French sentences. Relationships between the sentences are represented by connecting lines.

generate translations (Lepage and Denoual, 2005; Langlais et al., 2008; Dandapat et al., 2010).

A step further, Taillandier et al. (2020) proposed to fuse the direct approach with the indirect approach. See Figure 1(c). The translation output can be obtained by solving the following three analogical equations.

i am coming . : *j' arrive .* :: *i am eating an apple .* : *??*

he 's eating an apple . : *il est en train de manger une pomme .* :: *i am eating an apple .* : *??*

il est en train d' arriver . : *j' arrive .* :: *il est en train de manger une pomme .* : *??*

Here, our proposal is to use the Transformer model to establish direct connections between each input sentence and the output sentence, in contrast to the fusion approach of using three quadrilateral relations to obtain the translation result as shown in Figure 1(c). With our approach, the input sentence information is better synthesized to generate the target translation as illustrated in Figure 1(d). The use of multiple attention is expected to enhance the translation accuracy of the results.

3 Multi-sentence construction Transformer architecture

We propose a novel Transformer structure that allows for multiple sentences to be inputted simultaneously, compared to the vanilla Transformer's single-sentence input. Concretely, this multi-sentence construction Transformer architecture takes seven sentences A, B, C, D, A', B', C' as input to generate the target translation D'. Rather than concatenating them into a single input, we employ seven distinct inputs, which allows each individual input to compute attention with the output.

3.1 Structure of the decoder

The vanilla Transformer's decoder only receives two inputs to establish their connection: the sequence of vector representation of the source sentence from the encoder and the sequence of the target sentence. As an initial step towards building our multi-sentence construction Transformer architecture, a new decoder that can accommodate three inputs is designed in Figure 2.

To learn the relationship with the upper decoder, we add an extra layer of cross-attention after self-attention. This layer calculates the attention between the upper decoder's output and the target sentence, enabling the computation of attention to each input with the target output and establishing a connection. As a result, we create a decoder with three inputs for follow-up use.

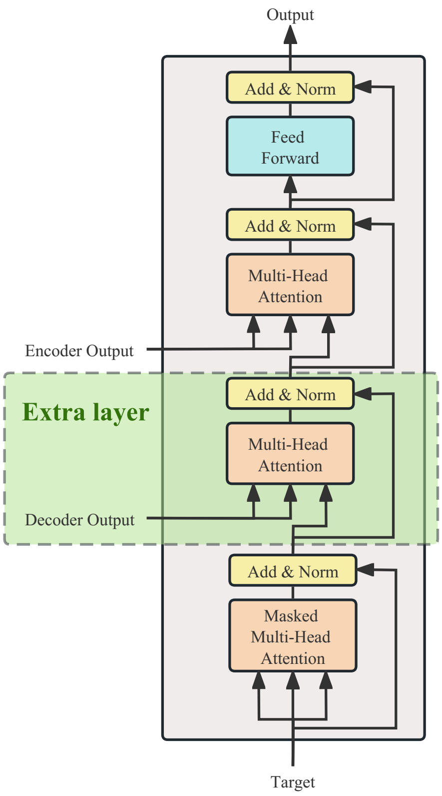


Figure 2: Structure of the decoder

3.2 Architecture for EBMT Transformer

Figure 3 illustrates the contrast between our proposed model architecture and the vanilla Transformer. Our proposal transforms the Transformer's single input into several independent encoder-decoder pairs. With multiple decoder layers overlaying each other, our EBMT Transformer can automatically encode the semantic information of the input sequence and use it to generate the appropriate target sequence.

Therefore, in the multi-sentence construction Transformer architecture for EBMT by analogy:

- All the encoders have the same structure as the vanilla Transformer's encoder, but each encoder has a weight specific to the corresponding input.

- Except for *decoder_A* which has the same structure as the vanilla Transformer's decoder (a two-input decoder), the other six decoders are the three-input decoders introduced in Section 3.1.

4 Datasets and metrics

4.1 Datasets

We use experimental data obtained directly from the bilingual analogy dataset developed by Taillandier et al. (2020) for comparison. For the task of multilingual machine translation, it will be necessary to create a multilingual analogy dataset.

4.1.1 Bilingual dataset

All sentences in the bilingual analogy dataset (Taillandier et al., 2020) are from Tatoeba[1]. The dataset is randomly divided into a training set (80%), validation set (10%), and test set (10%) by the number of analogies. As shown in Table 1, the average sentence length is approximately 5 words. Table 1 also counts the number of unique sentences contained in the dataset. Despite the fact that the whole dataset contains 239,594 analogies between sentences, it only contains 8,867 English sentences and 10,437 French sentences without repetition.

4.1.2 Multilingual dataset

To produce an analogy dataset for multiple languages, we extract analogies from the Tatoeba corpus using the Nlg package[2] (Fam and Lepage, 2018). Tatoeba is a collection of sentences in over 100 languages. In this work, we use English, French, German, Chinese, Japanese and Russian. Thus, we construct a multilingual dataset in six languages with 7,099 analogies and divide it into 80%, 10%, 10%.

Table 2 shows the statistics of the extracted multilingual dataset. In particular, each language has approximately 1,700 unique sentences. When considering the sentence length on the word level, Japanese has the longest average length and Russian has the shortest one.

4.2 Evaluation metrics

We automatically assess experimental results by comparing the translation output to the reference sentence in the test set. We use the three metrics listed below.

BLEU (Bilingual Evaluation Understudy) evaluates the similarity between the translated and reference sentences (Papineni et al., 2002). It features a 0 to 100 scale. The closer the translation output

[1] https://tatoeba.org
[2] http://lepage-lab.ips.waseda.ac.jp/media/filer_public/64/52/64528717-c3ce-4617-8208-c1fb70cf1442/nlg-v321.zip

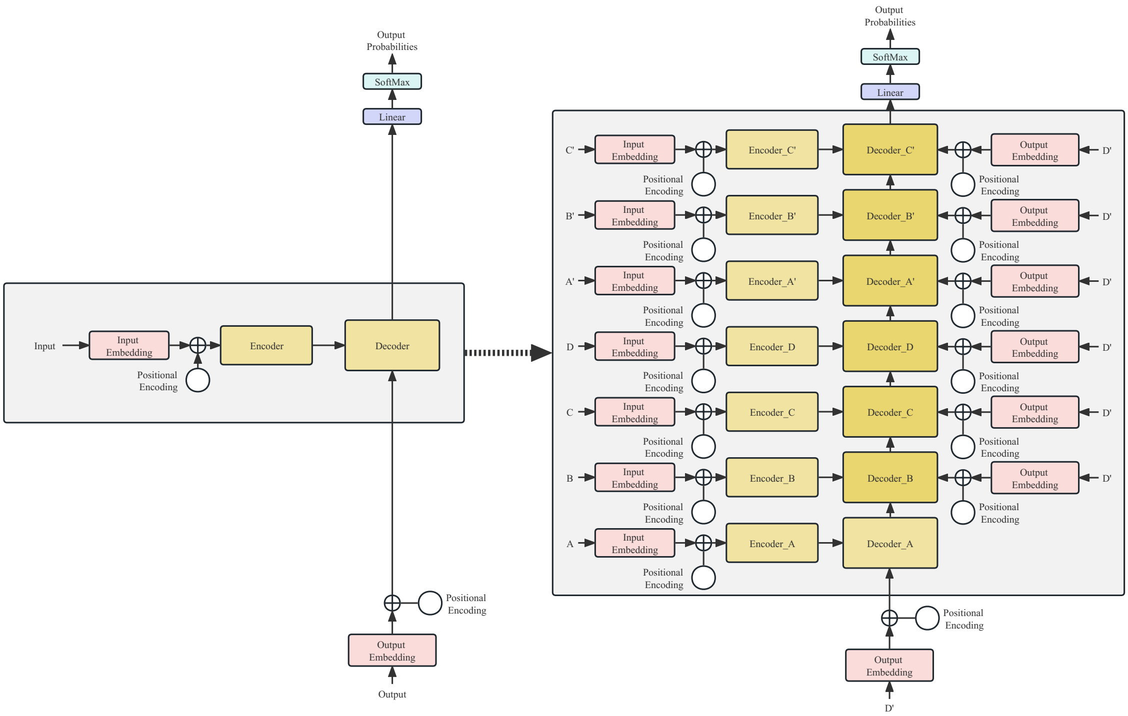


Figure 3: Model architecture: on the left, the vanilla Transformer, on the right, our EBMT Transformer.

Table 1: Statistics of the bilingual dataset

Dataset	Analogies	# of unique sentences		words/sentence		characters/sentence	
		English	French	English	French	English	French
train	191,676	8,867	10,437	5.50±1.45	5.66±1.57	21.31±6.04	24.54±7.33
valid	23,959	7,734	8,868	5.49±1.45	5.65±1.56	21.27±6.01	24.48±7.30
test	23,959	7,768	8,955	5.51±1.46	5.66±1.57	21.35±6.06	24.53±7.33

Table 2: Statistics of the multilingual dataset

Dataset	Analogies	Language	# of unique sentences	words/sentence	characters/sentence
train	5,679	English	1,746	5.65±1.50	21.39±6.41
		French	1,676	5.75±1.82	24.78±8.44
		German	1,665	5.17±1.34	24.14±7.68
		Chinese	1,628	4.95±1.33	11.35±3.08
		Japanese	1,662	6.90±2.42	17.17±5.82
		Russian	1,664	4.50±1.40	20.25±7.46
valid	710	English	956	5.61±1.45	21.13±6.20
		French	927	5.73±1.75	24.51±8.14
		German	928	5.14±1.29	23.88±7.51
		Chinese	915	4.91±1.30	11.22±2.99
		Japanese	916	6.91±2.35	17.17±5.69
		Russian	925	4.50±1.37	20.31±7.39
test	710	English	946	5.67±1.53	21.40±6.37
		French	915	5.74±1.82	24.63±8.43
		German	917	5.18±1.40	24.08±7.83
		Chinese	900	4.92±1.33	11.30±3.04
		Japanese	912	6.92±2.42	17.18±5.84
		Russian	916	4.49±1.38	20.11±7.40

is to the reference sentence, the higher the BLEU score is. We use SacreBLEU[3] (Post, 2018).

Accuracy refers to the percentage of translation results where the model outputs are identical to the reference sentences. This value can be expressed as the ratio of the number of identical results, denoted by n, to the total number of references, denoted by m. Mathematically, it can be represented as Accuracy $= n/m$.

Levenshtein edit distance (Levenshtein, 1966) is defined as the minimum number of edit operations (insertions, deletions or substitutions) required to transform one string into another. We evaluate the results using two units: word and character. A smaller edit distance indicates better results.

5 Experiments and analysis

To evaluate the performance of our proposed model, we compare its translation results to those of other methods. For the bilingual translation task from English to French, we use OpenNMT[4] (Klein et al., 2017) and the method proposed by Taillandier et al. (2020) as baselines. For the multilingual translation task across six languages, we use OpenNMT only. The parameter settings for OpenNMT and our proposal are detailed in Appendix A.

5.1 Bilingual translation task

Table 3 shows the translation results of various methods on the bilingual dataset mentioned in Section 4.1.1. Our proposed EBMT Transformer achieved a BLEU score of 98.8, outperforming OpenNMT's 90.3 and Taillandier et al. (2020)'s 94.7. Additionally, our model outperformed the baselines in terms of accuracy and edit distance metrics, demonstrating the stability of the results. Therefore, the multi-sentence construction Transformer architecture clearly provides a substantial improvement on this task.

Appendix B provides an error case for translation into French. Our proposed method faces challenges when it comes to accurately incorporating punctuation marks during the inference process.

5.2 Multilingual translation task

For multilingual translation across six languages, a total of $C_2^6 \times 2 = 30$ models need to be trained for each translation direction. The complete results are attached in Appendix C. Figures 4 and 5 present the BLEU score and accuracy of multilingual translation across six languages, respectively.

As shown in Figure 4(a), all OpenNMT models achieved a BLEU score of over 75. This is impressive given that OpenNMT typically requires a large amount of training data to achieve good results. However, as discussed in Section 4.1.2, the multilingual dataset used in this experiment only contains a total of 7,099 analogies, indicating that the dataset is very particular. We further observe that when English, French and Russian are the target language, the results are generally better than for other languages.

After comparing the BLEU score and accuracy in Figures 4 and 5, it can be concluded that the EBMT Transformer outperforms OpenNMT for all six languages. Although both methods have high translation performance, this is likely due to the fact that the languages involved do not have a large vocabulary and the sentences are short. The BLEU scores for Chinese as the target language are lower than those of other languages. This is mainly because Chinese has the lowest average number of characters per sentence, which results in a lower BLEU score calculation.

6 Conclusion

We proposed a multi-sentence construction Transformer architecture model to implement EBMT by analogy. Our proposal outperformed the two baselines on the bilingual dataset, achieving a BLEU score of 98.8 and an accuracy of 96.5. Additionally, for the multilingual translation task across six languages, our proposed method produced significantly better results than OpenNMT.

Limitations

Note that the used datasets are relatively easy ones. This raises questions about the generalizability of our proposed model when used in a real EBMT by analogy setting where retrieval of analogies from an input sentence should be taken into consideration. Future research will explore this issue using more complex datasets.

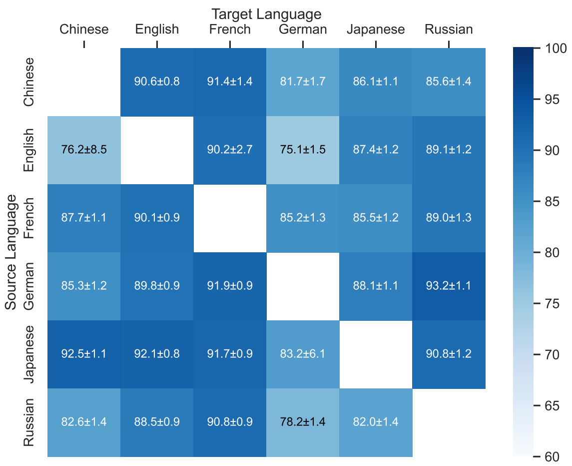
Acknowledgements

The results reported here were obtained in part under a JSPS grant, Kakenhi Kiban C, n° 18K11447 entitled "Self-explainable and fast-to-train example-based machine translation using neural networks."

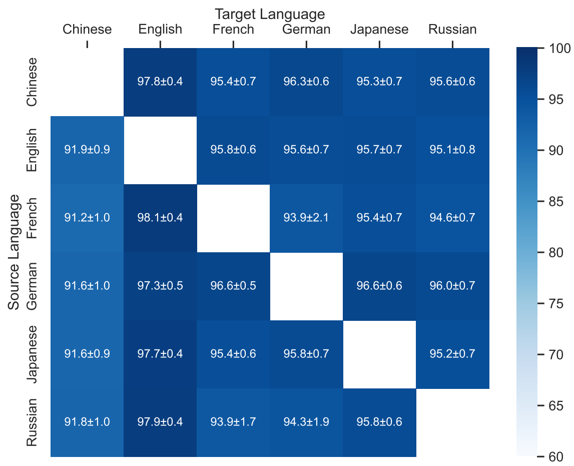
[3] https://github.com/mjpost/sacrebleu
[4] https://opennmt.net/

Table 3: Translation results of different methods on the bilingual dataset (en → fr)

Method	BLEU	Accuracy	Edit distance	
			in word	in char.
OpenNMT	90.3	82.7	0.5	1.0
(Taillandier et al., 2020)	94.7	90.2	0.2	0.6
EBMT Transformer	**98.8**	**96.5**	**0.1**	**0.2**

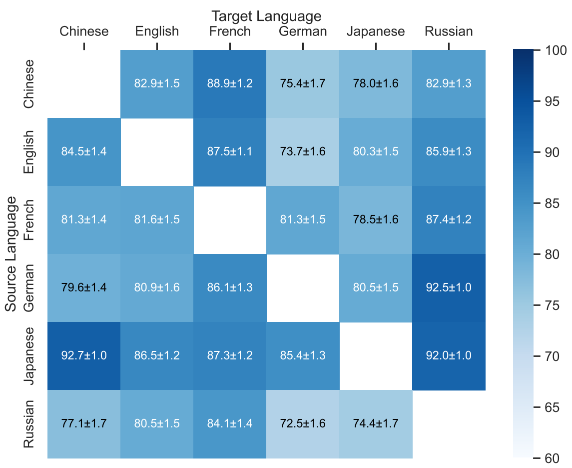


(a) OpenNMT

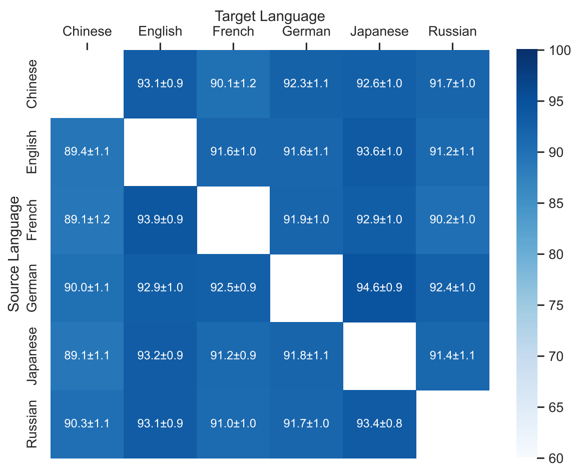


(b) EBMT Transformer

Figure 4: BLEU scores of multilingual translation across six languages



(a) OpenNMT



(b) EBMT Transformer

Figure 5: Accuracy of multilingual translation across six languages

References

Sandipan Dandapat, Sara Morrissey, Sudip Kumar Naskar, and Harold Somers. 2010. Mitigating problems in analogy-based ebmt with smt and vice versa: a case study with named entity transliteration. In *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation*, pages 365–372.

Rashel Fam and Yves Lepage. 2018. Tools for the production of analogical grids and a resource of n-gram analogical grids in 11 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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.

Philippe Langlais, François Yvon, and Pierre Zweigenbaum. 2008. Analogical translation of medical words in different languages. In *Advances in Natural Language Processing: 6th International Conference, GoTAL 2008 Gothenburg, Sweden, August 25-27, 2008 Proceedings*, pages 284–295. Springer.

Yves Lepage and Etienne Denoual. 2005. Purest ever example-based machine translation: Detailed presentation and assessment. *Machine translation*, 19(3/4):251–282.

V.I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics-doklady*, 10(8):707–710.

Makoto Nagao. 1984. A framework of a mechanical translation between japanese and english by analogy principle. *Artificial and human intelligence*, pages 351–354.

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.

Valentin Taillandier, Liyan Wang, and Yves Lepage. 2020. Réseaux de neurones pour la résolution d'analogies entre phrases en traduction automatique par l'exemple. In *6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 2 : Traitement Automatique des Langues Naturelles*, pages 108–121, Nancy, France. ATALA.

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 Experimental setup

Encoder&Decoder	
Type	Transformer
Embedding dimension	512
Number of layers	6
Number of heads	8
Size of feedforward layer	2048
Optimizer	Adam
Learning rate	1.0

Table 4: Parameter settings for OpenNMT

Encoder&Decoder	
Embedding dimension	512
Number of layers	1
Number of heads	8
Size of feedforward layer	2048
Optimizer	Adam
Learning rate	0.0001
Dropout	0.1
Max length	80

Table 5: Parameter settings for our proposal

B Error case in bilingual translation task

Table 6: Error case in bilingual translation task

IDs	Sentences
A	you're the love of my life .
B	you're such a jerk .
C	he's the love of my life .
D	he's such a jerk .
A'	tu es l'amour de ma vie .
B'	tu es un de ces pauvres types !
C'	c'est l'amour de ma vie .
Output	c'est un de ces pauvres .
Reference	c'est un de ces pauvres types !

C Results of multilingual translation task

See next pages.

Table 7: Result of multilingual translation task with OpenNMT

Source Language	Target Language	BLEU	Accuracy	Edit distance	
				in word	in char.
English	French	90.2 ± 2.7	87.5 ± 1.1	0.5 ± 0.2	1.9 ± 0.6
English	German	75.1 ± 1.5	73.7 ± 1.6	1.0 ± 0.1	4.3 ± 0.3
English	Chinese	76.2 ± 8.5	84.5 ± 1.4	1.1 ± 0.5	2.1 ± 1.0
English	Japanese	87.4 ± 1.2	80.3 ± 1.5	0.9 ± 0.1	1.8 ± 0.2
English	Russian	89.1 ± 1.2	85.9 ± 1.3	0.3 ± 0.1	1.6 ± 0.2
French	English	90.1 ± 0.9	81.6 ± 1.5	0.5 ± 0.1	1.7 ± 0.2
French	German	85.2 ± 1.3	81.3 ± 1.5	0.6 ± 0.1	2.8 ± 0.3
French	Chinese	87.7 ± 1.1	81.3 ± 1.4	0.4 ± 0.1	1.0 ± 0.1
French	Japanese	85.5 ± 1.2	78.5 ± 1.6	1.0 ± 0.1	2.1 ± 0.2
French	Russian	89.0 ± 1.3	87.4 ± 1.2	0.3 ± 0.0	1.5 ± 0.2
German	English	89.8 ± 0.9	80.9 ± 1.6	0.5 ± 0.1	1.6 ± 0.2
German	French	91.9 ± 0.9	86.1 ± 1.3	0.5 ± 0.1	1.8 ± 0.2
German	Chinese	85.3 ± 1.2	79.6 ± 1.4	0.5 ± 0.1	1.1 ± 0.1
German	Japanese	88.1 ± 1.1	80.5 ± 1.5	0.8 ± 0.1	1.7 ± 0.2
German	Russian	93.2 ± 1.1	92.5 ± 1.0	0.2 ± 0.0	0.9 ± 0.1
Chinese	English	90.6 ± 0.8	82.9 ± 1.5	0.5 ± 0.1	1.6 ± 0.2
Chinese	French	91.4 ± 1.4	88.9 ± 1.2	0.4 ± 0.1	1.6 ± 0.2
Chinese	German	81.7 ± 1.7	75.4 ± 1.7	0.8 ± 0.1	3.8 ± 0.4
Chinese	Japanese	86.1 ± 1.1	78.0 ± 1.6	0.9 ± 0.1	1.9 ± 0.2
Chinese	Russian	85.6 ± 1.4	82.9 ± 1.3	0.5 ± 0.1	2.0 ± 0.2
Japanese	English	92.1 ± 0.8	86.5 ± 1.2	0.4 ± 0.1	1.3 ± 0.2
Japanese	French	91.7 ± 0.9	87.3 ± 1.2	0.4 ± 0.1	1.7 ± 0.2
Japanese	German	83.2 ± 6.1	85.4 ± 1.3	0.8 ± 0.4	3.3 ± 1.6
Japanese	Chinese	92.5 ± 1.1	92.7 ± 1.0	0.2 ± 0.0	0.5 ± 0.1
Japanese	Russian	90.8 ± 1.2	92.0 ± 1.0	0.2 ± 0.0	1.1 ± 0.1
Russian	English	88.5 ± 0.9	80.5 ± 1.5	0.6 ± 0.1	1.9 ± 0.2
Russian	French	90.8 ± 0.9	84.1 ± 1.4	0.5 ± 0.1	2.0 ± 0.2
Russian	German	78.2 ± 1.4	72.5 ± 1.6	0.8 ± 0.1	3.8 ± 0.3
Russian	Chinese	82.6 ± 1.4	77.1 ± 1.7	0.7 ± 0.1	1.4 ± 0.1
Russian	Japanese	82.0 ± 1.4	74.4 ± 1.7	1.2 ± 0.1	2.4 ± 0.2

Table 8: Result of multilingual translation task with EBMT Transformer

Source Language	Target Language	BLEU	Accuracy	Edit distance in word	Edit distance in char.
English	French	95.8 ± 0.6	91.6 ± 1.0	0.2 ± 0.0	0.8 ± 0.1
English	German	95.6 ± 0.7	91.6 ± 1.1	0.2 ± 0.0	0.8 ± 0.1
English	Chinese	91.9 ± 0.9	89.4 ± 1.1	0.3 ± 0.0	0.4 ± 0.1
English	Japanese	95.7 ± 0.7	93.6 ± 1.0	0.3 ± 0.1	0.6 ± 0.1
English	Russian	95.1 ± 0.8	91.2 ± 1.1	0.2 ± 0.0	0.8 ± 0.1
French	English	98.1 ± 0.4	93.9 ± 0.9	0.1 ± 0.0	0.3 ± 0.1
French	German	93.9 ± 2.1	91.9 ± 1.0	0.3 ± 0.1	1.2 ± 0.4
French	Chinese	91.2 ± 1.0	89.1 ± 1.2	0.3 ± 0.0	0.4 ± 0.1
French	Japanese	95.4 ± 0.7	92.9 ± 1.0	0.3 ± 0.1	0.6 ± 0.1
French	Russian	94.6 ± 0.7	90.2 ± 1.0	0.2 ± 0.0	0.9 ± 0.1
German	English	97.3 ± 0.5	92.9 ± 1.0	0.1 ± 0.0	0.5 ± 0.1
German	French	96.6 ± 0.5	92.5 ± 0.9	0.2 ± 0.0	0.6 ± 0.1
German	Chinese	91.6 ± 1.0	90.0 ± 1.1	0.3 ± 0.0	0.3 ± 0.1
German	Japanese	96.6 ± 0.6	94.6 ± 0.9	0.2 ± 0.1	0.5 ± 0.1
German	Russian	96.0 ± 0.7	92.4 ± 1.0	0.2 ± 0.0	0.7 ± 0.1
Chinese	English	97.8 ± 0.4	93.1 ± 0.9	0.1 ± 0.0	0.4 ± 0.1
Chinese	French	95.4 ± 0.7	90.1 ± 1.2	0.3 ± 0.1	0.9 ± 0.1
Chinese	German	96.3 ± 0.6	92.3 ± 1.1	0.2 ± 0.0	0.6 ± 0.1
Chinese	Japanese	95.3 ± 0.7	92.6 ± 1.0	0.3 ± 0.1	0.7 ± 0.1
Chinese	Russian	95.6 ± 0.6	91.7 ± 1.0	0.2 ± 0.0	0.7 ± 0.1
Japanese	English	97.7 ± 0.4	93.2 ± 0.9	0.1 ± 0.0	0.4 ± 0.1
Japanese	French	95.4 ± 0.6	91.2 ± 0.9	0.3 ± 0.1	0.9 ± 0.1
Japanese	German	95.8 ± 0.7	91.8 ± 1.1	0.2 ± 0.0	0.8 ± 0.1
Japanese	Chinese	91.6 ± 0.9	89.1 ± 1.1	0.3 ± 0.0	0.4 ± 0.1
Japanese	Russian	95.2 ± 0.7	91.4 ± 1.1	0.2 ± 0.0	0.8 ± 0.1
Russian	English	97.9 ± 0.4	93.1 ± 0.9	0.1 ± 0.0	0.4 ± 0.1
Russian	French	93.9 ± 1.7	91.0 ± 1.0	0.3 ± 0.1	1.1 ± 0.3
Russian	German	94.3 ± 1.9	91.7 ± 1.0	0.3 ± 0.1	1.1 ± 0.4
Russian	Chinese	91.8 ± 1.0	90.3 ± 1.1	0.3 ± 0.0	0.3 ± 0.1
Russian	Japanese	95.8 ± 0.6	93.4 ± 0.8	0.3 ± 0.1	0.5 ± 0.1

Reconstruct to Retrieve:
Identifying Interesting News in a Cross-Lingual Setting

Boshko Koloski
Jožef Stefan Institute
Jožef Stefan IPS
boshko.koloski@ijs.si

Blaž Škrlj
Jožef Stefan Institute
Jožef Stefan IPS
blaz.skrlj@ijs.si

Nada Lavrač
Jožef Stefan Institute
nada.lavrac@ijs.si

Senja Pollak
Jožef Stefan Institute
senja.pollak@ijs.si

Abstract

An important and resource-intensive task in journalism is retrieving relevant foreign news and its adaptation for local readers. Given the vast amount of foreign articles published and the limited number of journalists available to evaluate their interestingness, this task can be particularly challenging, especially when dealing with smaller languages and countries. In this work, we propose a novel method for large-scale retrieval of potentially translation-worthy articles based on an auto-encoder neural network trained on a limited corpus of relevant foreign news. We hypothesize that the representations of interesting news can be reconstructed very well by an auto-encoder, while irrelevant news would have less adequate reconstructions since they are not used for training the network. Specifically, we focus on extracting articles from the Latvian media for Estonian news media houses. It is worth noting that the available corpora for this task are particularly limited, which adds an extra layer of difficulty to our approach. To evaluate the proposed method, we rely on manual evaluation by an Estonian journalist at Ekspress Meedia and automatic evaluation on a gold standard test set.

1 Introduction

Media houses often report relevant foreign news and adapt them to the local readership. With the ever-rising number of published articles and the limited number of people retrieving and curating the stories, the task becomes harder for media houses. The media houses often need to allocate scarce resources available, such as translators of specific languages and journalists, to curate and adapt the stories. In this work, we propose an approach that, given a handful of articles in a given language (Estonian), automatically suggests a set of potentially interesting news in a chosen foreign language (Latvian), employing a deep auto-encoder network to reconstruct and retrieve the relevant foreign articles. The task of identifying foreign interesting news is defined by the Estonian media house, interested in retreival of Latvian articles. For example, articles covering international politics (e.g. American elections) are not interesting, as the Estonian house would have other sources for these news. Also, many local articles are not interesting, as they are irrelevant for Estonians. However, very specific articles are of their interest, including the ones, covering Estonians in Latvia, topics relevant to Estonian readership (e.g. discussion on electronic scooters, doping affairs. These topics are however not predefined, but there is a small dataset of retrieved interesting news. The ratio of interesting to non-interesting news is very small, suggesting the task to be considered as imbalanced classification. In many imbalanced classification tasks (such as phishing detection (Douzi et al., 2017), software defect prediction(Tong et al., 2018), wind-turbine (Roelofs et al., 2021) anomaly detection), auto-encoders models have been utilized due their ability to reconstruct subgroups of examples well. Zhang and Zhu (2020) used Wasserstein auto-encoders for document retrieval, while Liou et al. (2014) used word-based auto-encoders for document retrieval.

This work extends the previous work on interesting cross-border news retrieval by Koloski et al. (2021), where the authors define a custom metric – SNIR *(Seed news of interest ratio)*. First, the method embeds both the set of *interesting* articles and the set of candidate articles into a multilingual space (Conneau et al., 2020). Next, the SNIR score of each *candidate* is calculated as the fraction of

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 81–89
September 11–12, 2023. ©2023 Association for Computational Linguistics

the *interesting* articles in a neighborhood of m articles. This metric follows the nearest-neighbor-based approach, where they check the ratio of interesting versus non-interesting news in the neighborhood for a given article. If the ratio is bigger than a given threshold, then the article is considered as interesting and thus relevant for translation and adaptation. They define an article as interesting if it is highly relevant to the Estonian readership at the time of publishing. To our knowledge, this is the only related work for the addressed task. We extend this work by proposing a novel method, as well as by proposing an automated evaluation setting for our task.

In the rest of this paper, Section 2 analyzes related work, Section 3 describes the data used, followed by the explanation of the proposed method in Section 4, and its evaluation in Section 5. Conclusions and further work are presented in Section 6.

2 Related work

In the field of journalism, one of the crucial responsibilities is to search for and gather captivating news stories from neighboring countries. Recent research by Asim et al. (2019) examines the use of ontologies, a type of language technology, in the domain of news retrieval. According to their findings, ontologies are primarily used for semantic search in news retrieval systems. Additionally, the collaboration between translation and journalism is essential in the process of news retrieval (Conway and Davier, 2019; Valdeón, 2020). Machine translation plays a significant role in automatically converting news stories in different languages to a language that is familiar to the news media curator (Utiyama and Isahara, 2003; Kumano et al., 2002; Eck et al., 2004; Bielsa and Bassnett, 2008; Almahasees, 2018).

Large Language Models (LLMs) are currently at the forefront of the field of machine translation. There are mainly two types of LLMs: autoregressive and autoencoding. Autoregressive models generate text by predicting the next word in a sequence given all the previous words. Examples of autoregressive models include GPT-3 (Brown et al., 2020) - model based on the Casual Language Modeling (CLM) task and BERT (Devlin et al., 2018) - model trained with the Masked Language Modeling (MLM) objective. Autoencoding models, on the other hand, are trained to reconstruct the original input given a corrupted version of it. These models learn to represent the input in a compact form that captures the most important information. Examples of autoencoding models include T5 (Raffel et al., 2020) and BART (Lewis et al., 2020). Both types of LLMs are highly effective for a wide range of tasks in interest to journalism: genre identification (Kuzman et al., 2023), text classification (Sun et al., 2019; Koloski et al., 2022a), sentiment analysis (Shirsat et al., 2017; Godbole et al., 2007; Bautin et al., 2008; Balahur et al., 2013; Keivandarian and Carvalho, 2023), machine translation (Zhu et al., 2020; Clinchant et al., 2019; Weng et al., 2020; Hendy et al., 2023), keyword extraction (Martinc et al., 2022; Koloski et al., 2022b) and more. Moreover, multilingual variants of these models (such as XLMR(Conneau et al., 2020)) have been developed to support multiple languages, making them even more useful for cross-lingual NLP tasks.

Autoencoder networks have found widespread use for input retrieval via reconstruction in various domains. For instance, Lu et al. (2021) developed a Siamese autoencoder for dense text retrieval, Xu et al. (2021) benchmarked a network consisting of an autoencoder and a generative adversarial network for zero-shot cross-modal retrieval, reporting promising results. Additionally, Ma et al. (2022) investigated the effect of contrastive pre-training for dense retrieval via autoencoder networks and achieved highly favorable outcomes. In this paper, we apply autoencoders to discover *interesting* news by reconstructing documents. We define *Interesting news* (based on prior work (Koloski et al., 2021)) as news that readers relate to and originates from foreign countries.

3 Data

The data used in this work consists of Estonian and Latvian articles (published in the period between *01.01.2018* until *01.12.2019*) by media houses belonging to the Ekspress Meedia Group. We used the following corpora from the EMBEDDIA news archives data set (Pollak et al., 2021).

- The collection of **Estonian** news articles from the archives of Ekspress Meedia, resulting in 17,148 articles
- The collection of **Latvian** news articles published by the DELFI portal - a Latvian subsidiary of the Ekspress Meedia Group. We

used the data before 1.12.2019 for training (29,178 articles) and the data after for testing (1,339 articles). We split the data in this manner to assess the method's ability to generalize over unseen news and events for the future.

- The set of **21 Latvian** news, consisting of articles published (between 01.01.2019 and 31.12.2019) in the Latvian journal and identified by an Estonian journalist as being interesting for the Estonian public. We also dispose of their aligned **Estonian** counterparts, which are the news that was published in the Estonian newspaper after translation and adaptation.

4 Method

4.1 Data Acquisition

Automated Acquisition of Estonian Ground Truth Our method follows the work by (Koloski et al., 2021) consists of two steps. In the first step, we use exact string matching to extract Estonian articles that mention Latvian Delfi[1] *(Läti Delfi, Lati Delfi, Delfi.lv)* in the article body text as a source of news. The hypothesis is that these articles were identified as significant for translation/adaptation from their original Latvian counterparts. In this manner, we acquired 100 Estonian articles, and we denote them as $\textbf{Estonian}_{\textbf{ground}}$.

4.1.1 Cross-Lingual Mapping

We hypothesize that the potentially interesting Latvian news are the ones that are in a joint multilingual space of Estonian and Latvian articles, gravitating closer to the surrounding of each article of the $\textbf{Estonian}_{\textbf{ground}}$. To do so, we follow the (Zosa et al., 2020) methodology for extracting articles in a multilingual setting:

1. We use sentence-transformers (Reimers and Gurevych, 2019) *XLM-r-distilRoBERTa-base-paraphrase-v1* embeddings to embed the articles from $\text{Estonian}_{\text{ground}}$ and the $\text{Latvian}_{\text{train}}$ articles in a common, multilingual space.

2. For each article $E_i \in \text{Estonian}_{\text{ground}}$ collection, we select $k \in \{1, 100\}$ closest Latvian articles (based on the Euclidean distance, efficiently computed via a KD-tree (Bentley, 1975) structure), obtaining a collection of Latvian articles $LE_{i,k}$ for each article of the $\textbf{Estonian}_{\textbf{ground}}$ articles.

3. Finally, we join all of the sets $LE_{i,k}$ from the previous step, obtaining the final $\textbf{Latvian}_{\textbf{extracted@k}}$ - Latvian extracted set of articles.

At the end of this step, for a given k, we obtain a collection of training articles. The number of articles in the $\textbf{Latvian}_{\textbf{extracted@k}}$, for a chosen k is shown in Figure 2.

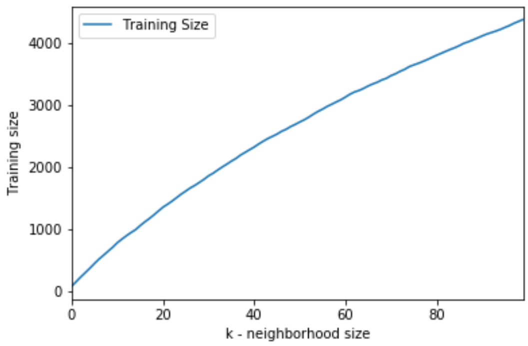


Figure 2: Distribution of articles for given k-neighborhood.

To evaluate the mapping, the Mean Reciprocal Rank (MRR) between the mappings of Estonian to Latvian articles, and vice-versa, were computed for the 21 pairs, where we obtained an average MRR of 66.67%. Even if the linking is incorrect, we assume that even when we do not retrieve the exact match, the articles in the identified neighborhood k still represent a neighborhood of potentially interesting source articles.

4.1.2 Validation Set of Manually Labeled Positive and Negative Examples

For positive examples, we used the 21 manually identified interesting Latvian news $21P$ (see Section 3). However, no negative examples were provided. Therefore, we extracted five random articles for every Latvian article in the $21P$ collection, obtaining a list of 105 articles. A journalist from Ekspress Media manually checked the list and identified 38 articles as unimportant for retrieval. We denote these articles as **NL**. We combined the 21 Latvian examples from the $21P$ collection with the 38 negative articles from the **NL** set, forming a validation set **V**.

[1] Delfi is one of the biggest news portals in Estonia and Latvia, many other media outlets (some of which contributed to the original dataset) often cite this source.

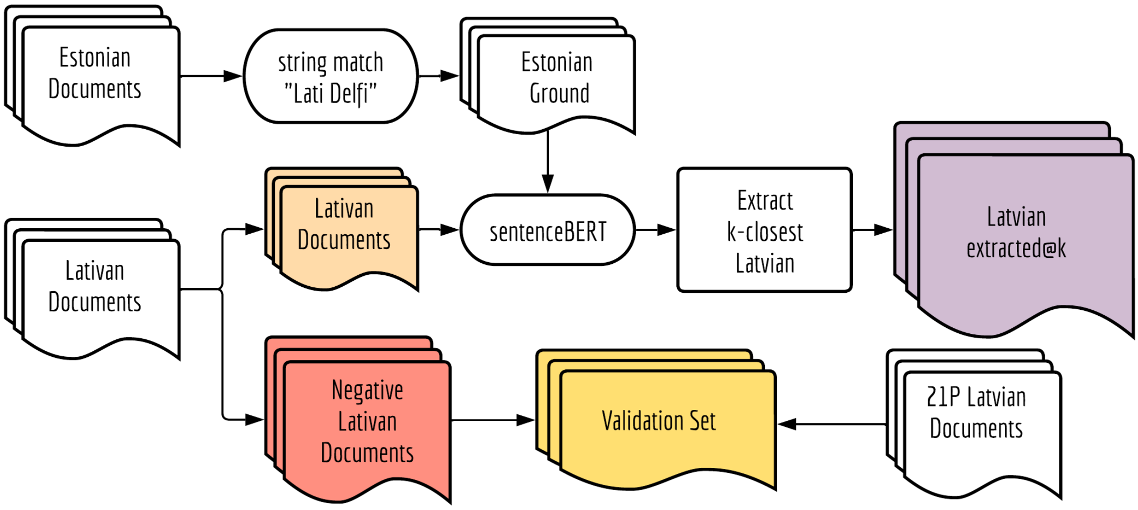


Figure 1: Summarization of our data-acquisition approach.

4.1.3 Experimental Data

We used the following experimental data sets, constructed as explained above:

- The training set $\textbf{Latvian}_{\textbf{extracted@k}}$ consisting of the mapped Latvian **k**-neighborhood articles obtained for every $\text{Estonian}_{\text{ground}}$ article. Figure 2 represents the distribution of articles per various k.
- The validation set **V** consists of 21 positive and 37 negative Latvian examples. The validation set was used to set the classification threshold and evaluate the auto-encoder network, as presented in Section 4.2.

4.2 Learning

We postulate that articles of interest share similar representation patterns. To investigate this hypothesis, we use a set of k Latvian articles from the $\text{Latvian}_{\text{extracted@k}}$ set to learn representations using deep auto-encoder network architectures. We experiment with several deep auto-encoder network architectures to identify the most effective approach. The core concept of the network is to take the original representation of an article, denoted by L_i, and encode it into a lower dimension, obtaining a compressed intermediate representation denoted by C_{L_i}. The encoder part of the network performs the encoding, while the decoder learns to reconstruct the code back to the original representation, yielding a reconstructed representation denoted by L_i^*. By learning these representations, we can better understand the common patterns shared by articles of interest and use this knowledge to improve our retrieval method.

4.2.1 Hyperparameters

We consider using two types of networks for our auto-encoder-based neural network: regularized and non-regularized. To embed the articles, we use the *XLM-r-distilRoBERTa-base-paraphrase-v1* model from sentence-transformers (Reimers and Gurevych, 2019), which converts them to 768-dimensional vectors that serve as input. Our encoder architecture has five layers with 512, 256, 128, 64, and 32 dimensions, while the decoder reverses the same architecture. We use the ReLU (Nair and Hinton, 2010) activation function between layers for all architectures. Figure 3 illustrates the architecture setup.

We optimize our network by using the Mean Squared Error between the reconstructed (L^*) and original (L) representations as the loss function, with the Adam optimizer (Kingma and Ba, 2014) and a learning rate of 0.001. We train for up to 1000 epochs and stop early if we don't improve the validation score in 10 consecutive epochs.

4.2.2 Classification Settings

The auto-encoder outputs the reconstructions of the original input and cannot be used directly for classification. However, in many imbalanced classifications (Zhang et al., 2016) and outlier detection (Chaurasia et al., 2020) problems, the auto-encoder

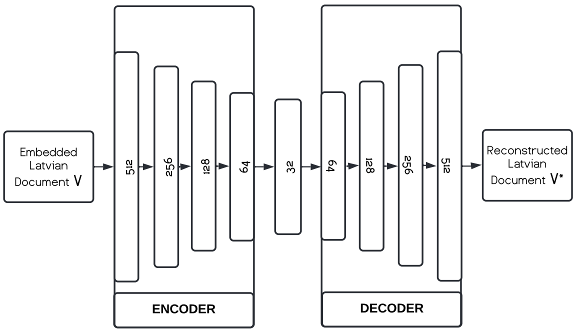


Figure 3: Architecture configuration. The encoder and decoder consist of the same architecture.

is used to prioritize outputs based on its reconstruction error (via thresholding). We use the following scoring function:

$$g(L^*, L, t) = \begin{cases} 1 & \text{cosineSimilarity}(L^*, L) \geq t; \\ 0 & \text{otherwise;} \end{cases}$$

where L^* is the reconstructed and, L is the original representation. The classification threshold is denoted by t. To classify an example after a network is trained, we first reconstruct it through the network and apply the classifying function g.

4.2.3 Threshold Learning

In each learning epoch, we reconstruct the validation examples from the set V, which includes 21 positive and 37 negative gold standard examples. This produces a list of reconstructed articles, denoted by $V*$. Then, we measure the reconstruction errors and create a list of errors $R_{k,e}$, where k denotes the population size and e denotes the epoch.

To determine the classification threshold, we search the grid with a step size of 0.01, denoted by stepRange $= [\min(R_{k,e}), \max(R_{k,e})]$. We test each step value as a potential threshold value t. We apply the classifying function g with t and compute the weighted F1-score of the classified reconstructions. We select the t value that yields the optimal F1 score. Formally, we choose t such that:

$$\underset{t \in \text{stepRange}}{\text{argmax}} \left[\text{F1-score}\Big((g(V*, V, t), \text{gold-standard}) \Big) \right]$$

This process enables us to determine the classification threshold that maximizes the F1-score for the reconstructed articles in the validation set, thus providing an effective means for classifying the reconstructed articles.

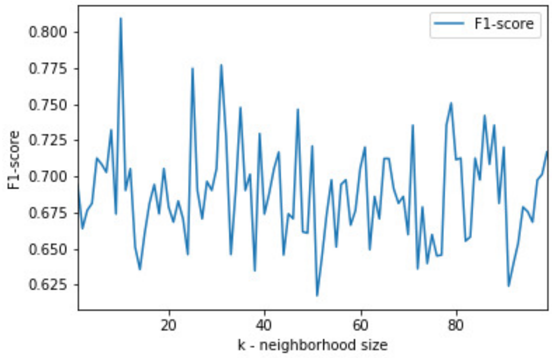


Figure 4: Distribution of F1-scores for the optimal threshold parameter at given k-neighborhood.

The non-regularized *Model32* outperformed the regularized model. Table 1 lists the parameters and evaluations. The model achieved[2] weighted F1-score of 0.81, recall-score of 0.8103, precision score of 0.8087 and accuracy of 0.8103. Figure 4 represents the effect of the training size to the validation score. The confusion matrix for the best-performing validation is listed in Figure 5.

Although the approach necessitates the utilization of negative examples for acquiring the optimal threshold, its purpose is to "regularize" the auto-encoder within the latent space. This ensures that the method doesn't retain specific events in memory but instead contributes to a more effective regularization process.

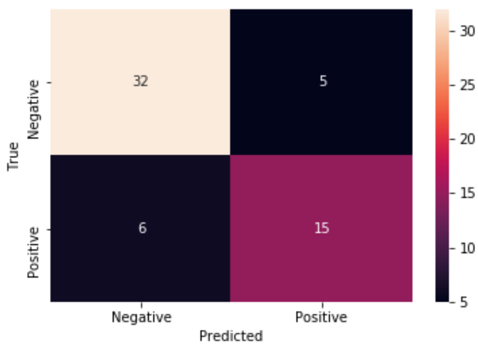


Figure 5: Confusion matrix of the best-performing validation.

5 Evaluation

We evaluate the method in two scenarios, manual and automated. In both systems, we use the test-

[2]True-negatives = 32, False-negatives = 6, False-positives = 5, True-positives = 15

Name	Type	train-size	k-neigh	threshold	epoch	F1-score	Yes	Maybe	No
Model32	Non-regularized	712	10	0.6035	11	**0.8093**	2	2	6
Model32D	Regularized	1951	32	0.5961	5	0.7608	0	2	8
Baseline	Randomized	x	x	x	x	0.4967	0	0	10

Table 1: Summary of the settings and evaluations for the best-performing networks. The optimal threshold is shown in the *threshold* column, followed by the number of epochs trained in the *epoch* column. Finally, the F1-score represents the validation score, followed by the manual evaluations (*YES*/*MAYBE*/*NO*). The human evaluator carried out the evaluations.

ing data for retrieving the top-ranked articles as interesting and relevant.

5.1 Manual Evaluation

We retrieved the top 10 articles (20 in total) in two different network settings and compared them to a baseline (10 randomly chosen articles). To assess the task, we use two different network configurations and a baseline:

- **Model32**, non-regularized network
- **Model32D**, regularized network
- **Baseline**, a random selection of articles sent to the evaluator. We consider the majority *NO* in the F1-score.

The testing data set consists of 1339 articles, which are input to our network and the reconstruction error is measured. The top-10 reconstructed articles with the smallest reconstruction errors are considered potentially interesting and sent to a journalist expert.

A journalist at Ekspress Media manually evaluates the retrieved articles in the categories introduced in (Koloski et al., 2021), i.e., *YES* - the article is definitely relevant, *MAYBE* - the article is relevant to some extent and *NO* - the article is of no relevance. The results are described in Table 1. The journalist found two articles of definitive relevance and 2 of possible relevance for retrieval in the best settings. Given that the problem is difficult, i.e., retrieving very special articles from a large set of all articles, the results still indicate that for Model32, 40% of the articles are potentially interesting. This is slightly lower than the results of (Koloski et al., 2021), wherein the best setting, one more article, was labeled as MAYBE. Of the 30 articles we sent for evaluation to the human evaluator, two were chosen as interesting, four as MAYBE, and the remaining as not interesting.

5.2 Automated Evaluation

This subsection demonstrates that our method performs better than random article retrieval. We first create a test set comprising of $21P$ labeled Latvian articles and the Latvian$_{\text{test}}$ set for automatic evaluation. Next, we run an auto-encoder and measure the reconstruction errors without applying threshold classification. Then, we sort the articles by their reconstruction scores and search for the $21P$ relevant articles while retrieving the top-k articles. We use $Model32$ to calculate the recall@k to assess the performance, treating the $21P$ articles as the gold standard. We also establish a baseline using random scoring of articles, where we randomly shuffle the articles in the test set and conduct 10^6 random evaluations. As shown in Figure 6, the results suggest that our method outperforms the random retrieval method for identifying interesting articles for Estonian readers. Therefore, our method shows promise for further investigation and improvement in the future.

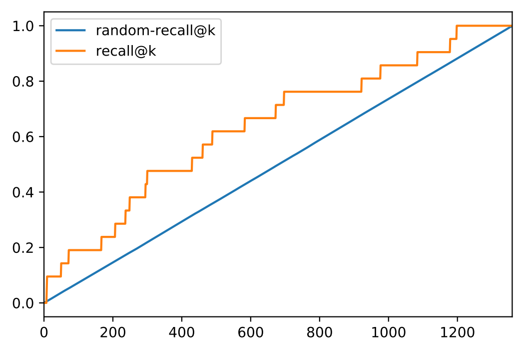


Figure 6: Recall comparison of the distributions. X-axis showcases the number of documents k, while y-axis shows the cumulative recall ($recall@k$).

6 Conclusion and further work

In this work, we have developed an auto-encoder-based approach for detecting and retrieving cross-border news. The method is trained unsupervised, given news datasets in two languages, relevant and non-relevant articles, and potential media houses hot words. The approach is shown to retrieve articles with 40% relevance, as evaluated manually by a media expert, and outperforms random-based approaches through recall@k evaluation.

For further work, we suggest exploring the significance of certain topics and keywords in a given time window, hypothesizing that story/topic relevance is time-dependent. We also propose exploring a term-matching approach that considers named entities and keyword matching to rank the relevance of an article. Lastly, we suggest investigating how combining the SNIR and auto-encoder as a weighted rank score could improve retrieval quality. To improve the relevance of retrieved articles, future work could explore the use of user feedback and relevance feedback mechanisms (such as RLHF). By incorporating user preferences and feedback, the system may be able to better tailor its results to the needs and interests of individual users.

Availability

The code required to replicate the experiments can be found at the following link: https://github.com/bkolosk1/reconstruct_to_retrieve.

Acknowledgements

The authors acknowledge the financial support from the Slovenian Research Agency for research core funding (No. P2-0103) and the project Computer-assisted multilingual news discourse analysis with contextual embeddings (CANDAS, J6-2581). A Young Researcher Grant PR-12394 supported the work of the first author.

Limitations

While the method is promising, it has limitations, such as a small evaluation set and the fact that tokens are not masked during retrieval, which may require retraining on a temporal basis. These limitations may affect the generalizability of the method to larger datasets and other languages. In addition, further investigation into the significance of topics and keywords in a given time window and using a term-matching approach could also enhance the method's effectiveness. Additionally, our investigation is limited to comparing the method solely against unsupervised methodologies, which restricts the scope of our work and opens up possibilities for further improvement.

Ethics Statement

The authors have used only existing datasets and do not identify any elements for ethical considerations.

References

Zakaryia Mustafa Almahasees. 2018. Assessment of google and microsoft bing translation of journalistic texts. *International Journal of Languages, Literature and Linguistics*, 4(3):231–235.

Muhammad Nabeel Asim, Muhammad Wasim, Muhammad Usman Ghani Khan, Nasir Mahmood, and Waqar Mahmood. 2019. The use of ontology in retrieval: A study on textual, multilingual, and multimedia retrieval. *IEEE Access*, 7:21662–21686.

Alexandra Balahur, Ralf Steinberger, Mijail Kabadjov, Vanni Zavarella, Erik Van Der Goot, Matina Halkia, Bruno Pouliquen, and Jenya Belyaeva. 2013. Sentiment analysis in the news. *arXiv preprint arXiv:1309.6202*.

Mikhail Bautin, Lohit Vijayarenu, and Steven Skiena. 2008. International sentiment analysis for news and blogs. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 2, pages 19–26.

Jon Louis Bentley. 1975. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517.

Esperança Bielsa and Susan Bassnett. 2008. *Translation in global news*. Routledge.

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.

Siddharth Chaurasia, Sagar Goyal, and Manish Rajput. 2020. Outlier detection using autoencoder ensembles: A robust unsupervised approach. In *2020 International Conference on Contemporary Computing and Applications (IC3A)*, pages 76–80.

Stephane Clinchant, Kweon Woo Jung, and Vassilina Nikoulina. 2019. On the use of bert for neural machine translation. *arXiv preprint arXiv:1909.12744*.

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.

Kyle Conway and Lucile Davier. 2019. Journalism and translation in the era of convergence. *Journalism and Translation in the Era of Convergence*, pages 1–217.

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*.

Samira Douzi, Meryem Amar, and Bouabid El Ouahidi. 2017. Advanced phishing filter using autoencoder and denoising autoencoder. In *Proceedings of the International Conference on Big Data and Internet of Thing*, BDIOT2017, page 125–129, New York, NY, USA. Association for Computing Machinery.

Matthias Eck, Stephan Vogel, and Alex Waibel. 2004. Language model adaptation for statistical machine translation based on information retrieval. In *LREC*.

Namrata Godbole, Manja Srinivasaiah, and Steven Skiena. 2007. Large-scale sentiment analysis for news and blogs. *Icwsm*, 7(21):219–222.

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are gpt models at machine translation? a comprehensive evaluation. *arXiv preprint arXiv:2302.09210*.

Neda Keivandarian and Marco Carvalho. 2023. A survey on sentiment classification methods and challenges. In *The International FLAIRS Conference Proceedings*, volume 36.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Boshko Koloski, Timen Stepišnik Perdih, Marko Robnik-Šikonja, Senja Pollak, and Blaž Škrlj. 2022a. Knowledge graph informed fake news classification via heterogeneous representation ensembles. *Neurocomputing*, 496:208–226.

Boshko Koloski, Senja Pollak, Blaž Škrlj, and Matej Martinc. 2022b. Out of thin air: Is zero-shot cross-lingual keyword detection better than unsupervised? In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 400–409, Marseille, France. European Language Resources Association.

Boshko Koloski, Elaine Zosa, Timen Stepišnik-Perdih, Blaž Škrlj, Tarmo Paju, and Senja Pollak. 2021. Interesting cross-border news discovery using cross-lingual article linking and document similarity. In *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, pages 116–120.

Tadashi Kumano, Isao Goto, Hideki Tanaka, Noriyoshi Uratani, and Terumasa Ehara. 2002. A translation aid system by retrieving bilingual news database. *Systems and Computers in Japan*, 33(8):19–29.

Taja Kuzman, Nikola Ljubešić, and Igor Mozetič. 2023. Chatgpt: Beginning of an end of manual annotation? use case of automatic genre identification. *arXiv preprint arXiv:2303.03953*.

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.

Cheng-Yuan Liou, Wei-Chen Cheng, Jiun-Wei Liou, and Daw-Ran Liou. 2014. Autoencoder for words. *Neurocomputing*, 139:84–96.

Shuqi Lu, Di He, Chenyan Xiong, Guolin Ke, Waleed Malik, Zhicheng Dou, Paul Bennett, Tie-Yan Liu, and Arnold Overwijk. 2021. Less is more: Pretrain a strong siamese encoder for dense text retrieval using a weak decoder. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2780–2791.

Xinyu Ma, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2022. A contrastive pre-training approach to discriminative autoencoder for dense retrieval. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4314–4318.

Matej Martinc, Blaž Škrlj, and Senja Pollak. 2022. Tnt-kid: Transformer-based neural tagger for keyword identification. *Natural Language Engineering*, 28(4):409–448.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Senja Pollak, Marko Robnik-Šikonja, Matthew Purver, Michele Boggia, Ravi Shekhar, Marko Pranjić, Salla Salmela, Ivar Krustok, Tarmo Paju, Carl-Gustav Linden, et al. 2021. Embeddia tools, datasets and challenges: Resources and hackathon contributions. In *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, pages 99–109.

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*, 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*. Association for Computational Linguistics.

Cyriana M.A. Roelofs, Marc-Alexander Lutz, Stefan Faulstich, and Stephan Vogt. 2021. Autoencoder-based anomaly root cause analysis for wind turbines. *Energy and AI*, 4:100065.

Vishal S Shirsat, Rajkumar S Jagdale, and SN Deshmukh. 2017. Document level sentiment analysis from news articles. In *2017 international conference on computing, Communication, Control and Automation (ICCUBEA)*, pages 1–4. IEEE.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*, pages 194–206. Springer.

Haonan Tong, Bin Liu, and Shihai Wang. 2018. Software defect prediction using stacked denoising autoencoders and two-stage ensemble learning. *Information and Software Technology*, 96:94–111.

Masao Utiyama and Hitoshi Isahara. 2003. Reliable measures for aligning japanese-english news articles and sentences. In *Proceedings of the 41st annual meeting of the association for computational linguistics*, pages 72–79.

Roberto A. Valdeón. 2020. On the interface between journalism and translation studies: A historical overview and suggestions for collaborative research. *Journalism Studies*, 21(12):1644–1661.

Rongxiang Weng, Heng Yu, Shujian Huang, Shanbo Cheng, and Weihua Luo. 2020. Acquiring knowledge from pre-trained model to neural machine translation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9266–9273.

Xing Xu, Jialin Tian, Kaiyi Lin, Huimin Lu, Jie Shao, and Heng Tao Shen. 2021. Zero-shot cross-modal retrieval by assembling autoencoder and generative adversarial network. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 17(1s):1–17.

Chenggang Zhang, Wei Gao, Jiazhi Song, and Jinqing Jiang. 2016. An imbalanced data classification algorithm of improved autoencoder neural network. In *2016 Eighth International Conference on Advanced Computational Intelligence (ICACI)*, pages 95–99.

Yifei Zhang and Hao Zhu. 2020. Discrete wasserstein autoencoders for document retrieval. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8159–8163.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. Incorporating bert into neural machine translation. *arXiv preprint arXiv:2002.06823*.

Elaine Zosa, Mark Granroth-Wilding, and Lidia Pivovarova. 2020. A comparison of unsupervised methods for ad hoc cross-lingual document retrieval. In *Proceedings of the workshop on Cross-Language Search and Summarization of Text and Speech (CLSSTS2020)*, pages 32–37, Marseille, France. European Language Resources Association.

Linguistic Pattern Analysis in the Climate Change-Related Tweets from UK and Nigeria

Ifeoluwa Wuraola

University of Hull, School of Computer Science, Hull, UK

i.a.wuraola-2021@hull.ac.uk

Nina Dethlefs

University of Hull, School of Computer Science, Hull, UK

n.dethlefs@hull.ac.uk

Daniel Marciniak

University of Hull, School of Criminology, Sociology and Policing, Hull, UK

d.f.marciniak@hull.ac.uk

Abstract

To understand the global trends of human opinion on climate change in specific geographical areas, this research proposes a framework to analyse linguistic features and cultural differences in climate-related tweets. Our study combines transformer networks with linguistic feature analysis to address small dataset limitations and gain insights into cultural differences in tweets from the UK and Nigeria. Our study found that Nigerians use more leadership language and informal words in discussing climate change on Twitter compared to the UK, as these topics are treated as an issue of salience and urgency. In contrast, the UK's discourse about climate change on Twitter is characterised by using more formal, logical, and longer words per sentence compared to Nigeria. Also, we confirm the geographical identifiability of tweets through a classification task using DistilBERT, which achieves 83% of accuracy.

1 Introduction

The IPCC reported in 2022 that climate change is currently impacting all inhabited regions worldwide, with human activities contributing to many observed changes in physical and biological systems (IPCC, 2022). The electronic data produced by internet users during climate-related events can offer valuable insights into how different geographic areas perceive the risks associated with climate change (Vicari et al., 2019). However, intercultural dialogue and discourse are increasingly being studied in linguistics, as culture is seen as a fundamental aspect of human activity (Hong et al., 2003). One such area of study is conversation culturomics, which uses language analysis to understand human culture and can help conservationists respond to cultural trends while staying socially relevant. Previous research by Ladle et al. (2016) identified ways in which this analysis can be useful, such as assessing the cultural impact of conservation interventions and promoting public understanding.

Our research aims at contributing to the comprehension of the interactions that exist across the UK and Nigeria by analysing and identifying linguistic features that each group uses to communicate the climate change narrative and therefore gain insights into the factors that shape these opinions and identify areas where more education and conservation interventions are needed. This is relevant as previous research by Diehl et al. (2019) highlighted a focus on Anglophone culture in studying climate change and conceptions on social media. Therefore, this study's inclusion of African perspectives contributes to the overall understanding of cultural differences in climate-related discourse.

In this paper, we use linguistic feature analysis supported by transformer networks to enhance classification performance and generate insights on cultural differences in climate-related discourse. Seen as only roughly 2% of Twitter data is geo-tagged (Karami et al., 2021), our study is based on a small dataset, and we look to overcome the limitations of data sparsity when analysing specific cultures through the inclusion of linguistic and socio-cultural features.

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 90–97
September 11–12, 2023.

2 Related Works

Researchers have contributed to the field of NLP by providing a wide range of approaches and techniques for analysing and predicting sentiments. Recent research by Tyagi et al. (2020) proposed a framework to examine the conversations on climate change between two communities on Twitter, namely activists and sceptics. The framework compares users' hashtags, bot percentage, and messaging to understand the differences between the communities. The study found that sceptics' messages focused more on attacking personalities, while activists' messages aimed to call for action against climate change. In addition to polarising tweet users, social media have been used to analyse emotions in tweets. Loureiro and Alló, (2020) studied climate policy opinions on Twitter in the UK and Spain. Results show UK sentiment is more positive, with anticipation prevailing, while fear is dominant in Spain. Gender analysis also indicates higher male tweeting in both countries, yet Spain demonstrates a more balanced gender distribution. Also, Hannak et al. (2012) research investigated sentiment patterns in tweets, particularly weather and time's impact on aggregate sentiment, and evaluated how clearly the well-known individual patterns translate into population-wide patterns. Machine learning techniques with weather-correlated tweets shows aggregate sentiment follows distinct climate, temporal, and seasonal patterns.

Furthermore, Chen et al. (2019) used deep neural networks to detect climate change skeptics from tweet content and analyzed Twitter's climate change discussions and influencing factors over time. They created a neural network model with an 88% accuracy in identifying deniers. Extreme weather events and policy shifts were noted to influence public interest and attitudes toward climate change.

Studies that systematically explore intercultural differences are rare. Liu and Zhao, (2017) show that NGOs in China typically work to frame climate change within national Chinese context, to highlight the relevance and impact for all Chinese people. This common perspective is typical in collectivist cultures (Diehl et al., 2021). In contrast, African NGO outlets often frame climate change discourse in terms of increased agricultural hardships and social hardships and place an emphasis on educating citizens on the matter (Ford et al., 2015). Education is also a topic in other Asian countries, including Afghanistan, Bhutan, Kiribati, Nepal, and Tuvalu, where NGOs additionally work towards implementing adaptive strategies to climate change risks and increasing scientific scholarship (McGregor et al., 2018). All these studies are based on social science methodology and often rely on extensive manual annotation. In this paper we want to explore how linguistic features can overcome the burden of annotation. Also, Schäfer and Painter, (2021) contrast climate journalism in the global north and south. The authors find that while climate coverage has changed globally over the last decade to move increasingly online, there are fewer journalists who specialise specifically on climate in the global south, which impacts public information in those regions accordingly.

This paper proposes a new method to analyse linguistic features and cultural differences in climate-related tweets. Our study combines transformer networks with linguistic feature analysis to address small dataset limitations and gain insights into cultural differences. We conclude by highlighting the importance of understanding cultural differences, particularly in the Nigerian/African perspective, in climate discourse to facilitate effective action against climate change. The focus on the Nigerian/African perspective adds a novel contribution to the existing literature.

3 Methodology

3.1 Data Collection and Curation

Key words	Nigeria	UK
“climate change”	*“my believe is that climate change will eventually lead to lives extinction”*	*“net zero is a hoax that nobody is falling for manmade climate change is a lie and is impossible”*
“global warming”	*“climate change global warming is a scam tho”*	*“yes to solve global warming is not about carbon banks and not filling ones kettle society will have to take look at itself and decide what kind of future we our children and grandchildren will have”*

Table 1: Sample tweets across Nigeria and UK based on key words.

Twitter has provided a wealth of data for analysis with 1 billion monthly visitors and 313 millions active users (Li et al., 2013), including topics such as climate change. Therefore, the datasets for this research were gathered from the Twitter API between September 2010 to April 2023. Tweets were filtered based on the key words shown in Table 1.

A total of 81,507 tweets were collected for the analysis, comprising 44,071 from the UK and 37,436 from Nigeria. Also, only English-language tweets (since the English set is the focus of this project) were kept. These are identified by running the tweets through a langdetect algorithm (a model that identifies the language used in the text within the specified range).

3.2 LIWC

Our study uses Linguistic Inquiry and Word Count (LIWC-22) to perform linguistic analysis on geo-tagged data collected from Twitter. LIWC-22 is a software developed by Boyd et al. in 2022. It analyses word use within a text and calculates the percentage of word use for certain linguistic categories. A study in Indonesia had previously used LIWC to filter names from tribe, religion, and race and observed that the use of names is mostly followed by negative sentiments (Adi and Eka, 2021). This paper uses LIWC to investigate the linguistic differences in climate change discourse in English tweets from the UK and Nigeria. The LIWC analysis produced 124 variables of word categories, and the top 20 variables with the highest variance were selected for further analysis. In addition to linguistic features, valence (degree of positiveness or negativeness) was also calculated using VADER (Valence Aware Dictionary for sEntiment Reasoning), a sentiment analysis tool commonly used for analysing social media data (Hutto, 2022).

3.3 Classification Task

Our study proposes a transformer network to classify country distribution based on geo-tagged tweets to verify the geographical origin of the tweets. As fine-tuning BERT can be challenging due to its complex structure and parameters, we use DistilBERT created by Sanh et al. (2020), a compressed version with fewer parameters that is easier and faster to fine-tune with moderate resources. The study explores the use of DistilBERT in pre-training and fine-tuning to build the model.

4 Results and Discussion

4.1 Linguistics Feature Analysis across Nigeria and UK

For each tweet, we take the average of the valence and LIWC values of each of the words in the tweet

	Description	Nigeria	UK	p-value
Valence	Positiveness-negativeness	0.17	0.13	<0.0001
	LIWC variables			
Tone	Degree of positive (negative) tone	41.51	40.48	<0.0001
Authentic	Perceived honesty, genuineness	31.79	31.49	>0.0001
Clout	Language of leadership, status	58.14	31.490	<0.0001
Analytic thinking	Metric of logical, formal thinking	72.96	75.13	<0.0001
Linguistics	Linguistics dimensions	45.78	46.86	<0.0001
Pronoun	I, you, that, it	7.53	6.92	<0.0001
Determinants	the, at, that, my	9.05	9.51	<0.0001
WPS	Average words per sentence	28.88	31.46	<0.0001
WC	Total word count	28.88	31.46	<0.0001
Cognition	Psychological processes (is, was, but, are)	9.25	8.84	<0.0001
Social	Social processes	9.32	7.61	<0.0001

Big words	Percent words 7 letters or longer	27.61	28.10	<0.0001
Cogproc	Cognitive processes (but, not, if, or, know)	8.34	7.93	<0.0001
Perception	in, out, up, there	7.86	8.37	<0.0001
Verb	is, was, be, have	9.88	9.57	<0.0001
Dictionary words	Percent words captured by LIWC	64.69	64.76	>0.0001
Function	the, to, and, I	35.09	35.26	>0.0001
Adjective	more, very, other, new	5.34	6.37	<0.0001
Preposition	to, of, in, for	10.53	10.76	<0.0001
Allure	have, like, out, know	5.82	5.13	<0.0001

Table 2: Description of variables, average valence, LIWC scores, and p-value at country level.

text. The average of the variables was computed at the country level. Table 2 displays the valence and 20 selected variables for the study, with the highest value being highlighted, and presenting the distribution p-value for each. We observe the average valence of Nigerian tweets is higher (more positive) than the tweets from the UK.

4.2 Interpretation of Findings

According to the study, Nigerians tend to use more leadership language in climate change discourse on Twitter compared to the UK, which may be due to cultural and social norms. Nigerian culture places more emphasis on leadership and authority figures, given the country's political instability and history of leadership challenges in climate change and environmental degradation (Uyigue and Agho, 2007). As a result, climate-related tweets in Nigeria may be targeted toward government officials or policymakers, leading to the use of more leadership language to convey a sense of urgency and importance. Examples of tweets to support this claim are "*save niger delta environment from pollution extinction of sea foods activist pleads portharcourt the federal government and indeed the people of the niger delta have been urged to pull resources together amp save the region s environment from further pollution*" and "*climateaction requires significant investments by governments and businesses but climate inaction is vastly more expensive lets all unite to take climate actions now*". In addition to leadership, Nigeria's vulnerability to the impacts of climate change may be reflected in the use of emotional and informal language when discussing climate-related matters (Adelekan, 2010), making such issues more salient and urgent to them. Also, the study finds that Nigerian climate-related tweets have a higher frequency of verbs and pronouns compared to UK tweets. This may be because leadership is often expressed through specific linguistic features in text, such as the use of certain pronouns and verbs. (Tweets above use verbs such as 'save', 'take' which could be an attempt to convey a sense of urgency to the government). Furthermore, cognitive terms like "but", "if", etc, and social terms like "you", "we", "he", etc, are also higher in tweets from Nigeria compared to the UK.

The study finds that the discourse about climate change on Twitter in the UK is characterised by using 'bigger' and more formal/logical words, frequent use of adjectives and prepositions, and longer words per sentence compared to discourse from Nigeria. This difference could be attributed to the demographics of the authors of tweets, including NGOs and news outlets with more technical knowledge in climate science or related fields. This results in the use of specialised and technical lexicons, which is more evident in the UK than in Nigeria. Examples of tweets to support the claim are "*discovering different life perspectives could be just around the* *corner join climate solutions book club to make new friends from around the world and start exploring we read one book a month on climate change and then meet up on zoom to discuss https*" and "*human tiger conflicts seen to rise as migrants move into nepal national park conservationists raise concerns that the growing human presence in the chitwan district will pose additional challenges to conservation efforts https*" Additionally, the cultural emphasis on politeness and formality in the UK could also influence language use on social media (Sifianou, 1999). Another possible factor is that the urgency of climate change issues may not be as immediately felt or apparent in the UK as the nation is developed and her government is more proactive in addressing climate change policies compared to the Nigerian government, leading to a more detached and analytical discussion (O'Neill and Nicholson-Cole, 2009; Loureiro and Alló, 2020; Vu, 2020).

To further investigate individual differences in formal and informal language use, a cluster analysis was conducted on the dataset using the KMeans++ algorithms with 2 clusters, valence, and 20 LIWC variables as the experimental setup. The goal was to further explore formal-informal linguistic differences and observe whether clusters would automatically divide along this dimension. Specifically, we wanted to understand whether UK tweets were generally more formal than Nigeria tweets, or whether the type of user account that puts out a formal tweet is just more frequent in the UK than Nigeria. This is in line with research by Hopke and Hestres, (2018), who analyse the social media coverage during COP 21 (Paris) climate talks by different stakeholders. The authors show that while idiosyncrasies exist at a national level, pro-climate stakeholders, such as mainstream media outlets, NGOs, and prominent activists, showed notable similarities in the way they communicated about climate change and risks across countries and continents. Their analysis is manual and based on framing but seems in line with our linguistic findings. At a more detailed level, these findings imply that categorization of tweets using linguistic patterns and word use might assist environmental stakeholders in gaining knowledge of the target location to address scepticism about climate change and identify regions that require more education and advocacy.

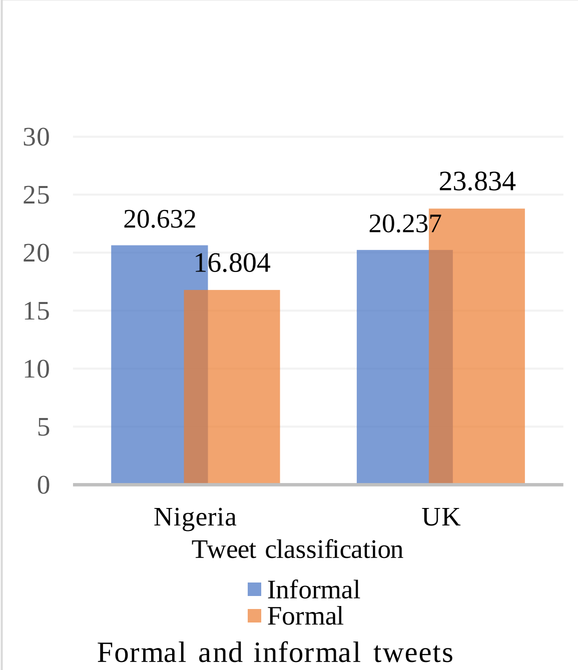


Figure 1: A figure showing Tweet classification across Nigeria and UK.

4.3 Experimental Results and Interpretation (classification tasks)

For our classification study, we randomly set the training size to 80% and we conduct experiments using 2 layers, 100 hidden units, dropout of 0.1, learning rate of 0.0001, Adam optimisation and 10 epochs. We achieved 83% of accuracy from the DistilBERT model when predicting the originating country of a tweet. We present qualitative analysis of a set of examples in Table 3. As can be seen, our classifier was able to correctly predict the country origin of 4 out of 5 non-geotagged tweets, indicating its potential use in identifying the origin of climate-related tweets. This can help policymakers in promptly addressing climate change disasters in specific regions. The model's predictions suggest that certain linguistic features may have played a role in predicting the country origin of tweets. As discussed in section 4.2, use of longer and bigger words per sentence is more common in the UK's climate-related discourse than in Nigeria, and this could have affected the model's ability to make accurate predictions. An example is the third non-geotagged tweet in Table 3, which had longer words per sentence and was wrongly predicted as originating from the UK. The accuracy of the study's predictions may have been influenced by factors such as dataset size, bias in training data,

Non-geotagged tweet	Actual label	Predicted label
'i m in love with nature'	Nigeria	Nigeria
'no you said there is no link and deforestation is solely a problem for overpopulation'	UK	UK
'revealed rampant deforestation of amazon driven by global greed for meat https t co'	UK	UK
'it is a very good season with a lot of rain so please plant any native flora it will be a great service to nature here you can see the benefits of trees biodiversity native'	Nigeria	UK
'what a nice gift from nature why bother it greeniewo savetheplanet gogreenie'	UK	UK

Table 3: Classification performance of non-geotagged tweet.

or other issues. A dataset including tweets from a wider variety of countries and regions could help to better train the model to recognize linguistic patterns that are specific to different cultural contexts. Also, testing the model on different types of tweets, such as those related to other environmental issues or different types of disasters, could help to determine whether the model's performance is specific to climate-related tweets or whether it can be applied more broadly. Therefore, further research could be done in those areas to optimise the model's accuracy.

5 Conclusion

The study employed linguistic feature analysis supported by transformer networks to investigate cultural differences in climate-related discourse between Nigerian and British tweets, aiming to identify trends in their respective lexicons. Through this approach, we were able to achieve a respective baseline classification performance and mitigate the limitations of working with small datasets. Our findings suggest that studying linguistic patterns and word use are crucial areas of research in socio-cultural analysis tasks, particularly in the classification of tweets based on their location. This is particularly useful in quickly addressing climate change disasters in specific geographic areas, as well as gauging public interest in climate change, and characterising discourse in different cultures even with limited data availability. Our experiments with DistilBERT on small data yielded promising results, with an accuracy of 83% in correctly classifying the country of origin for climate discourse tweets.

Based on these findings, we could recommend that those interested in identifying the country of origin of climate discourse tweets using linguistics patterns should focus on language that conveys a clear positive or negative sentiment and complex language in a way that is perceived as authoritative. Additionally, while the level of analytical thinking and linguistic complexity may also be important in predicting the country of origin of climate discourse tweets, they may not have as significant an impact as the overall sentiment and complexity of language. Future research will examine tweets in languages other than English as well as tweets from other countries. In addition to the linguistics pattern, we will look at the emotional dynamics surrounding climate change in these countries over an extended period.

Limitations

Although the present research identified patterns in the linguistic features of tweets, it only analysed English tweets. Therefore, the multilingualism of Twitter users should be considered to gain deeper insights into linguistic patterns and word use, as this can improve the analysis and prediction of such patterns. Also, given the scope of this research, it's important to acknowledge its limitation in definitively establishing whether the observed cultural disparities are attributed to climate change or general language differences. This opens the door for future investigations, potentially applying our methods to other disciplines for broader insights. Furthermore, given the potential limitation that geotagging may lead to a non-representative Twitter sample, our research's utilization of DistilBERT for training becomes pivotal in addressing this concern. Thus, the integration of non-geotagged tweets into the classifier presents a promising direction for future investigations, effectively addressing potential limitations.

Ethics Statement

The study followed the ACL Ethics Policy to ensure ethical and responsible conduct throughout the research process. We collected and analysed publicly available tweets, ensuring privacy and confidentiality of the Twitter users. Also, we avoid perpetuating stereotypes or biases and conducted the research in a respectful manner that aligned with cultural norms and values. Informed consent was not required since the data was publicly available, but we anonymized the data to protect individual users. The study uses appropriate statistical and computational methods and shared our findings transparently with the wider research community. We are committed to upholding ethical principles in their research.

Acknowledgments

The authors express gratitude to the Centre of Excellence for Data Science, Artificial Intelligence and Modelling (DAIM) for generously funding and enabling this research.

References

Ibidun O Adelekan. 2010. Vulnerability of poor urban coastal communities to flooding in Lagos, Nigeria. *Environment and Urbanization*, 22(2):433–450. https://doi.org/10.1177/0956247810380141.

Ryan Boyd, Ashwini Ashokkumar, Sarah Seraj, and James Pennebaker. 2022. *The Development and Psychometric Properties of LIWC-22*. https://doi.org/10.13140/RG.2.2.23890.43205.

Xingyu Chen, Lei Zou, and Bo Zhao. 2019. Detecting Climate Change Deniers on Twitter Using a Deep Neural Network. In *Proceedings of the 2019 11th International Conference on Machine Learning and Computing*, pages 204–210, New York, NY, USA. Association for Computing Machinery. https://doi.org/10.1145/3318299.3318382.

Trevor Diehl, Brigitte Huber, Homero Gil de Zúñiga, and James Liu. 2019. Social Media and Beliefs about Climate Change: A Cross-National Analysis of News Use, Political Ideology, and Trust in Science. *International Journal of Public Opinion Research*, 33. https://doi.org/10.1093/ijpor/edz040.

Trevor Diehl, Brigitte Huber, Homero Gil de Zúñiga, and James Liu. 2021. Social Media and Beliefs about Climate Change: A Cross-National Analysis of News Use, Political Ideology, and Trust in Science. *International Journal of Public Opinion Research*, 33(2):197–213. https://doi.org/10.1093/ijpor/edz040.

Brett Q. Ford, Julia O. Dmitrieva, Daniel Dmitrieva, Yulia Chentsova-Dutton, Igor Grossmann, Maya Tamir, Yukiko Uchida, Birgit Koopmann-Holm, Victoria A. Floerke, Meike Uhrig, Tatiana Bokhan, and Iris B. Mauss. 2015. Culture shapes whether the pursuit of happiness predicts higher or lower well-being. *Journal of Experimental Psychology: General*, 144(6):1053–1062. https://doi.org/10.1037/xge0000108.

Aniko Hannak, Eric Anderson, Lisa Feldman Barrett, Sune Lehmann, Alan Mislove, and Mirek Riedewald. 2012. Tweetin' in the Rain: Exploring Societal-Scale Effects of Weather on Mood. *Proceedings of the International AAAI Conference on Web and Social Media*, 6(1):479–482. https://doi.org/10.1609/icwsm.v6i1.14322.

Ying-Yi Hong, Veronica Benet-Martinez, Chi-Yue Chiu, and Michael W. Morris. 2003. Boundaries of Cultural Influence: Construct Activation as a Mechanism for Cultural Differences in Social Perception. *Journal of Cross-Cultural Psychology*, 34(4):453–464. https://doi.org/10.1177/0022022103034004005.

Jill Hopke and Luis Hestres. 2018. Visualizing the Paris Climate Talks on Twitter: Media and Climate Stakeholder Visual Social Media During COP21. *Social Media + Society*, 4:205630511878268. https://doi.org/10.1177/2056305118782687.

C. J. Hutto. 2022. cjhutto/vaderSentiment.

IPCC. 2022. The Sixth Assessment Report of the IPCC | UNFCCC.

Amir Karami, Rachana Redd Kadari, Lekha Panati, Siva Prasad Nooli, Harshini Bheemreddy, and Parisa Bozorgi. 2021. Analysis of Geotagging Behavior: Do Geotagged Users Represent the Twitter Population? *ISPRS International Journal of Geo-Information*, 10(6):373. https://doi.org/10.3390/ijgi10060373.

Richard J Ladle, Ricardo A Correia, Yuno Do, Gea-Jae Joo, Ana CM Malhado, Raphaël Proulx, Jean-Michel Roberge, and Paul Jepson. 2016. Conservation culturomics. *Frontiers in Ecology and the Environment*, 14(5):269–275. https://doi.org/10.1002/fee.1260.

Linna Li, M. Goodchild, and Bo Xu. 2013. Spatial, temporal, and socioeconomic patterns in the use of Twitter and Flickr. https://doi.org/10.1080/15230406.2013.777139.

John Liu and Bo Zhao. 2017. Who speaks for climate change in China? Evidence from Weibo. *Climatic Change*, 140. https://doi.org/10.1007/s10584-016-1883-y.

Maria L. Loureiro and Maria Alló. 2020. Sensing climate change and energy issues: Sentiment and emotion analysis with social media in the U.K. and Spain. *Energy Policy*, 143(C).

Ian McGregor, Hilary Yerbury, and Ahmed Shahid. 2018. The Voices of Local NGOs in Climate Change Issues: Examples from Climate Vulnerable Nations. *Cosmopolitan Civil Societies: An Interdisciplinary Journal*, 10:63–80. https://doi.org/10.5130/ccs.v10.i3.6019.

Saffron O'Neill and Sophie Nicholson-Cole. 2009. "Fear Won't Do It": Promoting Positive Engagement With Climate Change Through Visual and Iconic Representations. *Science Communication*, 30(3):355–379. https://doi.org/10.1177/1075547008329201.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108 [cs]. https://doi.org/10.48550/arXiv.1910.01108.

Mike Schäfer and James Painter. 2021. Climate journalism in a changing media ecosystem: Assessing the production of climate change-related news around the world. *Wiley Interdisciplinary Reviews: Climate Change*, 12. https://doi.org/10.1002/wcc.675.

Maria Sifianou. 1999. *Politeness Phenomena in England and Greece: A Cross-Cultural Perspective*. Oxford University Press. Google-Books-ID: hy1RCbiyWYkC.

Aman Tyagi, Matthew Babcock, Kathleen M. Carley, and Douglas C. Sicker. 2020. Polarizing Tweets on Climate Change. arXiv:2008.13054 [cs].

Etiosa Uyigue and Matthew Agho. 2007. Coping with Climate Change and Environmental Degradation in the Niger Delta of Southern Nigeria.

Rosa Vicari, Ioulia Tchiguirinskaia, Bruno Tisserand, and Daniel Schertzer. 2019. Climate risks, digital media, and big data: following communication trails to investigate urban communities' resilience. *Natural Hazards and Earth System Sciences*, 19(7):1485–1498. https://doi.org/10.5194/nhess-19-1485-2019.

Hong Vu. 2020. Social media and environmental activism: Framing climate change on Facebook by global NGOs. *Science Communication*, Accepted.

Nut-cracking Sledgehammers: Prioritizing Target Language Data over Bigger Language Models for Cross-Lingual Metaphor Detection

Jakob Schuster
Linguistic Data Science Lab
Ruhr University Bochum
Germany
jakob.schuster@ruhr-uni-bochum.de

Katja Markert
Institute of Computational Linguistics
Heidelberg University
Germany
markert@cl.uni-heidelberg.de

Abstract

In this work, we investigate cross-lingual methods for metaphor detection of adjective-noun phrases in three languages (English, German and Polish). We explore the potential of minimalistic neural networks supported by static embeddings as a light-weight alternative for large transformer-based language models. We measure performance in zero-shot experiments *without access to annotated target language data* and aim to find low-resource improvements for them by mainly focusing on a k-shot paradigm. Even by incorporating a small number of phrases from the target language, the gap in accuracy between our small networks and large transformer architectures can be bridged. Lastly, we suggest that the k-shot paradigm can even be applied to models using machine translation of training data.

1 Introduction

Metaphors are a phenomenon of figurative language where meaning about a more abstract concept is expressed by applying it to a more concrete domain. According to cognitive linguistic theories by Lakoff and Johnson (1980), they are systematic linguistic instantiations of so-called Conceptual Metaphors. An example of a conceptual metaphor is `EMOTION IS LIQUID`, which manifests in expressions such as *bubbly personality, his anger boiled over* and *overflowing joy*. Other definitions describe metaphors as novel usages of words, in which the semantic preference of the syntactic arguments is violated. As an example, *to eat* prefers animate subjects and edible objects. The metaphor *The job ate his confidence* violates this preference (Wilks, 1975). Previous studies show metaphors to make up a substantial portion of natural language[1] and heavily influence decision-making in discourse (Thibodeau and Boroditsky, 2011), making their detection a valuable topic in NLP. Since conceptual metaphors are based around semantic concepts and not words, they are shared throughout similar cultures and can sometimes be directly translated (*Er griff mein Argument an - He attacked my argument*). In other cases, the same conceptual metaphor might exist in two languages, but is lexicalized differently. While a direct translation of *Seine Stimmung war im Keller - His mood was in the basement* could most likely still be understood, a more conventional phrasing would be *His mood plummeted* or *He was feeling down*. Therefore, metaphor detection across different languages is an interesting topic worth exploring. However, most annotated metaphor resources center on English.

In this paper, we investigate the application of modern zero-shot methods without access to annotated target language data for cross-lingual metaphor detection of adjective-noun phrases in three different languages. We go on to soften the zero-shot limitation and measure how smaller feed-forward models can become competitive to transformer-based systems, by incorporating a small number of target language phrases into the training process. Lastly, we apply the same few-shot paradigm to improve models which use machine-translated data and discuss the results.

2 Related Work

Previous works about metaphor detection were mostly monolingual and supervised. They often leveraged additional resources, static word embeddings and in more recent experiments pre-trained transformer models (Wilks et al. (2013), Do Dinh and Gurevych (2016), Choi et al. (2021)). The latter is currently the most commonly used option. In a shared task about metaphor detection in 2020 (Leong et al., 2020), more than half of all participants used some variation of a transformer architecture. Recent concerns regarding the alternative

[1] The VUA Metaphor Corpus by Steen et al. (2010) annotates around 12% as metaphoric.

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 98–106
September 11–12, 2023. ©2023 Association for Computational Linguistics

usage of static word embeddings for metaphor detection were also voiced for theory based reasons (Maudslay and Teufel, 2022). Work addressing cross-lingual metaphor detection includes Tsvetkov et al. (2014), where semantic features and word vectors were used to transfer English metaphor knowledge about adjective-noun or verb-subject-object phrases into Spanish, Farsi and Russian, Schneider et al. (2022), an unsupervised approach for a transfer from German to middle high German based on self-trained fasttext embeddings (Grave et al., 2018) and Sanchez-Bayona and Agerri (2022) who present first zero-shot results between their newly created Spanish corpus and the English VUAMC by Steen et al. (2010) by using XLM-RoBERTa (Conneau et al., 2019), a transformer based multilingual language model.

This leaves many possible approaches to cross-lingual transfer-learning for metaphor detection unexplored. A common way to allow for this transfer in other tasks is machine translation to convert training data from the source language into the target language or vice versa (Eger et al., 2018). Joulin et al. (2018a) provide a lightweight alternative to bigger transformer models, by aligning static fasttext embeddings across 44 languages through a cross-domain similarity local scaling criterion. While multilingual models do work in zero-shot scenarios, Lauscher et al. (2020) show the benefit of shifting to a k-shot scenario, in which small target language datasets of size k are incorporated into training. Similarly, Keung et al. (2020a) present findings which support that using a development set in the target language can improve performance by preventing catastrophic forgetting of multilingual knowledge during training. More recently, Large Language Models such as ChatGPT[2], are used as zero-shot or few-shot in-context learning systems (Laskar et al. (2023), Yuan et al. (2023)). ChatGPT is a model of the GPT-3.5 or GPT-4 series, which is trained through a reinforcement learning from human feedback component (Christiano et al., 2017) and also possesses multilingual knowledge.

3 Task and Data

This paper focuses on binary classification of the metaphoricity of adjective-noun tuples, since this setup had the most available data in several languages. In these phrases, the metaphoric meaning can stem from the conceptual transfer of either the meaning of the adjective (*stale idea*) or the noun (*economic slump*). We collected corpora of labeled phrases big enough for both training and testing in English, German and Polish. A small sample can be seen in Table 1.

Example Inputs	Gold Label
wet towel, old man, ...	0
broken home, cultural barrier, ...	1

Table 1: Example classification schema for the metaphor detection task of adjective-noun phrases. 1 indicates a metaphorical and 0 a literal meaning.

	Size	%M	#adj	ppa
DE	1677	25.5	297	5.6
EN	1968	50.0	668	2.9
PL	2052	50.4	241	8.5

Table 2: Comparison of the annotated source datasets. By measuring the simple attributes share of metaphoric phrases (**%M**), number of adjective types (**#adj**) and phrases per adjective (**ppa**), we can show how the different strategies result in different distributions.

The English corpus (Tsvetkov et al., 2014) is balanced for both classes, and consists of metaphor annotations of the 1000 most common adjectives and their co-occuring nouns in the TenTen Web corpus.[3] It has been filtered to exclude phrases which without context can be interpreted literally and metaphorically (e.g. *drowning students*).

The German corpus (Sick, 2020) follows the same annotation procedure as Tsvetkov et al. and is extracted from the German deTenTen13[4] corpus. The resulting dataset is not balanced between classes, but rather reflects the distribution of metaphoric tokens in natural language. The Fleiss' κ (Fleiss, 1971) measuring inter-annotator agreement is 0.34. Since this is a low IIA, we filter the corpus and only include phrases for which at least 4 of the 5 annotators agreed.

The Polish corpus (Mykowiecka et al., 2018) is constructed by preparing a list of metaphorical phrases and enriching it with additional common phrases in the National Corpus of Polish (Przepiórkowski and Patejuk, 2014), using the same adjectives. After we removed phrases that were labeled as *Both* metaphorical and literal,

[2] https://chat.openai.com/

[3] https://www.sketchengine.eu/ententen-english-corpus/

[4] https://www.sketchengine.eu/detenten-german-corpus/

the corpus is almost perfectly balanced (1018 metaphors and 1034 literal phrases).

Due to the similar collection strategies, we can observe examples of the same conceptual metaphors being present in every corpus. (EMOTIONALLY INDIFFERENT IS COLD: *cold justice*, *kalte Grausamkeit* and *zimna kalkulacja*). A comparison of all sources can be seen in Table 2. To even out the differences in size, we trim every corpus down to the size of DE, while keeping the overhang in a separate set for later experiments. We then perform a 70:15:15 train, dev, test split, resulting in 1173 phrases for training and 252 each for developing and testing. Since we use our own test splits, we have no previous results from literature to compare against.

4 Experiments

In this section, we describe a series of binary classification experiments of our collected phrases. Each experiment described is conducted for all six possible combinations of training and test splits of our three available languages. We prioritize stability of our results over ideal hyperparameters and aim to ensure a fair comparison. Therefore, in all following experiments, we incorporate early stopping, learning rate warm-up and report the average result of ten majority vote ensembles with seven seeds each.

4.1 Upper Bound

Previous work for similar semantic tasks have shown big gaps in performance between cross-lingual and monolingual set-ups (Nozza, 2021; Hsu et al., 2019). As an approximation of an achievable upper-bound for our cross-lingual models, we first conduct monolingual experiments with language-dependent BERT variations[5] and light-weight, fully connected feed forward neural networks, using fasttext word embeddings[6].

4.2 Zero-shot Models

The cross-lingual zero-shot experiments of this section are defined by the absence of annotated target language examples in the training dataset. We compare models of three different categories for cross-lingual zero-shot metaphor detection. The first category consists of networks powered by aligned **fasttext** word embeddings by Joulin et al. (2018b). We train three additional variations of this architecture:

- **fasttext+TrTr** and **fasttext+TrTe**, with translations of the training data into the target language or the test data into the source language.[7]
- **fasttext+TarDev** which employs a development set in the target language as proposed by Keung et al. (2020b). Using a development set in the target language can enable a checkpoint selection that best suits the test data.

The second category encompasses the two multilingual pre-trained transformer models **MBERT** (Devlin et al., 2018) and **XLM-R**[8], which are fine-tuned on the source language for the classification.

The final category describes a set of experiments, utilizing **ChatGPT** [9] as a classifier via prompting:

- **ChatGPT** is not given any additional information.
- **ChatGPT+ex** is provided with 20 random examples from the source language's training split before (In-Context Few-shot Learning)
- **ChatGPT+MIP** is provided with the (translated) Metaphor Identification Procedure by Group (2007) and asked for corresponding annotations (In-Context Instruction Learning)

Example prompts for all three ChatGPT methods can be found in the Appendix.

4.3 k-shot for Fasttext Models

Just as proposed by Lauscher et al. (2020), in this series we relax the zero-shot limitation to explore an inexpensive approach of mitigating the gap between cross-lingual and monolingual performance. We incorporate k randomly sampled data points from the target language's training, development or overhang data into the training process of the fasttext baseline. This sample is different

[5] All BERT variations are finetuned with a learning rate of 2e-5 and Adam's epsilon of 1e-8 for 8 epochs

[6] All our fasttext networks consist of three hidden layers ($h_1 = 300$, $h_2 = 150$, $h_3 = 50$), with a dropout chance of 5% and are trained for 5 epochs.

[7] We use the neural machine translation *Amazon Translate*, provided by the *Amazon Web Service*. It has to be noted that using a big NMT service such as Amazon Translate adds a hidden compute to all related experiments.

[8] XLM-R is finetuned with a learning rate and Adam's epsilon of 10e-5 for 6 epochs

[9] GPT-3.5 model of the May 3, 16k version with a temperature of 0.05

	huggingface transformer model	BERT ACC	fasttext ACC
DE	german-bert	**80.5**	79.3
EN	bert-base	**88.9**	83.7
PL	dkleczek/bert-base-polish-uncased-v1	85.9	**86.9**

Table 3: For each language, the used monolingual BERT model from the huggingface model hub and the accuracies produced by said monolingual BERT's experiments and the monolingual experiments using fasttext word embeddings.

for each seven seed ensembles and each set of size k is a subset of another set of a larger k. We report results for $k \in \{0, 10, 25, 50, 100, 200, 1173\}$, where $k = 1173$ shows the maximum achievable effect by including the whole training split of the target language.

4.4 k-shot for Translated Train Models

Lastly, we examine if even transformer-based language models, which are supported by machine translation, can still benefit from the k-shot paradigm. For this, we finetune monolingual BERT models in the target language on the translated training data as our $k = 0$ baselines. We then add additional k phrases of the training, development and overhang split of the target language. These are untranslated and authentical data points. To contrast their effect to the addition of more translated training data, we use the test split of the source language dataset to also finetune models with additional k translated phrases.

5 Results

In this section, we present the results of the experiments we conducted. For all of them, we report the accuracy on the test sets.

5.1 Upper Bound

Table 3 contains the results of the monolingual experiments. Overall, the BERT baselines outperform the fasttext model and the German dataset yields the lowest accuracy. However, this mainly serves as a potential upper-bound for the upcoming cross-lingual experiments.

5.2 Zero-Shot Models

Table 4 displays the accuracy of all zero-shot models. Generally, accuracy of all zero-shot systems varies across language pairs and models, with the inclusion of the German dataset seemingly often leading to worse results. Across all systems and

	DE ->EN	DE ->PL	EN ->DE	EN ->PL	PL ->DE	PL ->EN	avg.
fasttext	48.2*	60.0	63.6	68.3	59.0	62.9	60.3
fasttext+TrTr	59.9	**65.8**	47.6*	**72.2**	**63.1**	68.6	62.8
fasttext+TrTe	44.4*	55.1	56.7	65.4	60.7	68.6	58.4
fasttext+TarDev	48.1*	59.9	65.4	68.2	59.5	62.3	60.5
XLM-R	59.3	60.8	62.5	67.3	49.5*	74.6	62.3
MBERT	60.2	63.0	**66.3**	66.8	49.4*	68.0	62.2
ChatGPT	57.1	63.1	62.6	63.1	62.6	57.1	60.9
ChatGPT+ex	56.7	63.1	55.1	68.6	53.1	48.0*	57.4
ChatGPT+MIP	**77.3**	65.0	57.9	65.9	56.0	**74.6**	**66.1**

Table 4: Report of all the zero-shot baseline systems for every available language pair and the average across all language pairs. For **ChatGPT**, there is no actual source language from which we transfer knowledge to a target language. Therefore, the results for the two source languages are always identical. We mark every model worse than a random baseline with *.

languages, **ChatGPT+MIP** performed the best and achieves an average accuracy of 67%. On average, the other transformer models were able to outperform the plain fasttext architecture, albeit not for every language pair. When utilizing machine translation however, the models with translated training data nullified the gap to the transformer models in almost every pair, while the models with translated test data became worse overall. How dependent this behaviour is on the used translation service was not examined. We also observe that the inclusion of a development set in the target language does not bring a notable improvement to our fasttext architecture. This could be due to the small training data size, where not enough meaningfully different checkpoints are available for choosing. It is important to mention that all three of the categories feature models which performed worse than a random baseline. Models based on ChatGPT also display peculiar behaviour, with the additional information through examples of a source language seemingly weakening its predictive power. As expected, a comparison of Table 3 and Table 4 shows that transfer-learning across languages leads to a strong drop in performance for this task.

5.3 k-shot for fasttext Models

Figure 1 displays heatmaps of the change in accuracy for all language pairs for rising k. Identically to the findings of Lauscher et al., we can observe a static incline of accuracy with rising k. Combinations that performed poorly in zero-shot rapidly improve, even for small values of k. On average, fasttext outperforms MBERT and XLM-R at $k = 25$ and even our best ChatGPT+MIT model for $k = 100$. Comparisons of $k = 200$ and $k = 1173$

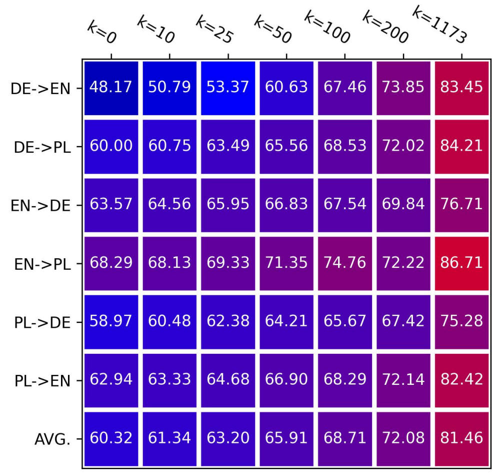


Figure 1: Heatmaps of accuracy across language pairs and the average across all pairs for rising values of k for vanilla fasttext models

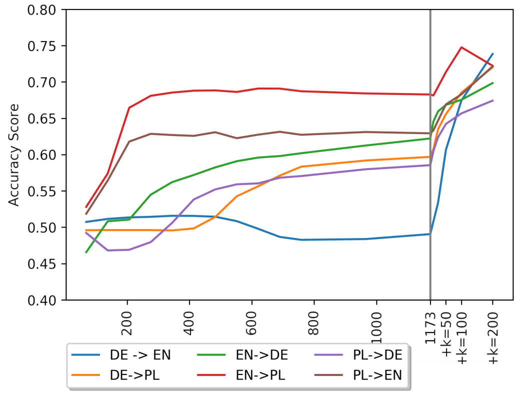


Figure 2: Learning curve of all language pairs for increasing source language training set size and additional k-shot data. The vertical line signifies the initiation of adding k-shot points to the complete training data set. Ten models area averaged for the reported accuracy score.

show that by only using 17% of the target language data, we can already obtain more than half of its potential increase in accuracy.

Another visual representation of the effectiveness of softening of the zero-shot limitation can be seen in Figure 2. The plot shows every available language pair's learning curve. By starting at an empty set and continuously adding data points of the source language to the training data set, it can be measured how much a model profits from more training data from that source. After adding the whole training data set, we then shift to further adding k-shot points. The vertical line indicates the point of this shift. It is evident that every model's learning curve slope gets steeper when switching

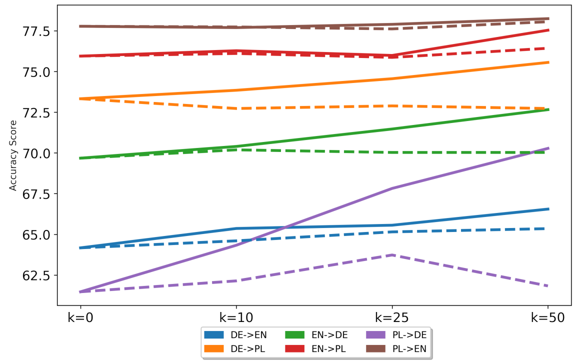


Figure 3: Comparison of accuracy across language pairs for rising values of k. Using the monolingual BERT models listed in Table 3 and additional translated training data (dashed line) or authentic data from the target corpus (solid line).

to the k-shot paradigm. This applies to pairs where the accuracy appears to plateau (EN $\rightarrow$ PL, PL $\rightarrow$ EN), to DE $\rightarrow$ EN, which seems to not improve at all and is outperformed by a random baseline, and to models which were past their stronger initial incline, but were still slightly improving.

5.4 k-shot for Translated Train Models

The zero-shot models of this experiment, while being not as light-weight due to the compute of BERT and the NMT, are roughly comparable to fasttext's $k = 100$ and $k = 200$ models in accuracy. The overall best models presented in this paper were obtained by this method for $k = 50$ for authentic k-shot, reaching an average accuracy of 73.47%. When comparing both methods in Figure 3, we can note that the models do not noticeably improve by additional translated training data. The same does not apply to the k-shot set of authentic data, where we observe a similar improvement to the k-shot experiments with fasttext - stronger improvements for models with worse performance in zero-shot.

6 Discussion

Impact of Machine Translation By looking at our translated data, we try to explain why the translation based zero-shot BERT experiments benefited more from the translations than the fasttext baselines. By our choice of method, we end up with translations of individual data points where the two word adjective-noun pair structure is lost (*warmer Milchschokoladenton* to *warm milk chocolate tone*, *crushed stone* to *Schotter*). By automatically POS-tagging the translated test data with spaCy (Honnibal and Montani (2017)), we measure these devia-

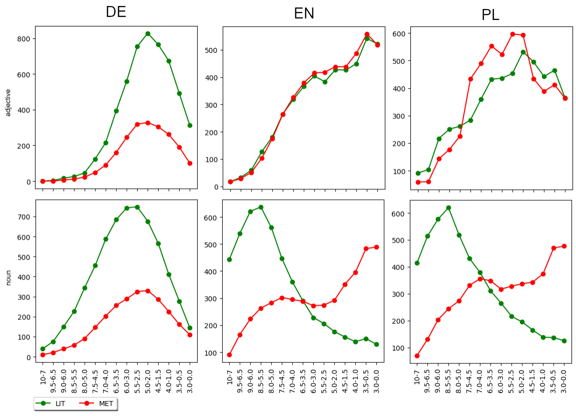


Figure 4: Distribution of abstractness/concreteness from the ratings provided by (Köper and Schulte im Walde, 2017) for the different corpora, separated into adjectives (top row) and nouns (bottom row). Words are rated on a scale of 0 to 10. Lower scores are given to abstract words (*irresponsibly*), higher to concrete words (*razor blade*).

tions in syntactic structure. Depending on language pair and direction, they make up between 7.5% and 38.4% of the translations. In general, length difference of the translations is less of a problem for the scalable transformer models, than for our 600 dimensional, fixed length neural network. This can be circumvented by using recurrent neural networks. Analogue to our findings of preferring simple models to large ones for this task, the same could possibly apply to the translation methods, since statistical or dictionary based methods could lead to less deviation in syntax and therefore to better results. This deviation can also explain how our models profit more from the authentic k-shot data, since it better represents the test data. We also observed the NMT already implicitly performing metaphor detection to better lexicalize concepts in the target language (*schwieriger Spagat* to *difficult balancing act* instead of a less conventional *difficult split* for the conceptual mapping `LIFE IS A SEQUENCE OF MOTION`).

Performance Difference Between Languages In order to try and explain the differences in performance for the individual language pairs, we investigated the semantic composition of the corpora. Using abstractness/concreteness ratings from Schulte im Walde (2022), we display the distribution of abstractness for the adjectives and nouns of our datasets in Figure 4. The DE corpus differs heavily, by having similar distributions of abstractness for metaphoric and literal words. In comparison, EN and PL contain more concrete nouns in literal and abstract nouns in metaphoric phrases. This is more in line with work by Turney et al. (2011), Tsvetkov et al. (2013) and Schulte im Walde (2022), where abstractness served as a classification feature and can serve as an indicator for the lower performance on the German test set.

Impact of k-shot Selection To gain insight into the effect of the selection of the k datapoints, we look at the performance of individual ensemble seeds with different k-shot sets. We investigate the intuitive connection between the seed's performance and the coverage of the test set adjectives by the k-shot data and show an exemplary scatter plot for our fasttext model and EN $\rightarrow$ PL in Figure 5. While larger values of k lead to a better performance and also naturally to a higher coverage of adjectives, when looking the distribution inside a cluster of k, there seems to be no strong connection. This makes the k-shot paradigm robust, since no knowledge about the word content of the test dataset is therefore needed. The plot also shows the k-shot data to improve both the detection of metaphors and the detection of literals. It is worth noting that the variance in performance appears to be higher for smaller values of k, with some poor performing outliers, while higher values of k produce more stable results.

Multiple efforts have been made to enhance the selection of k-shot data, similar to Lauschers selection based on length. Experiments based on attributes such as class label, frequency, distance of the data points in the vector space or other small handcrafted feature vectors were all unreliable and too dependent on the language pair and k. However, based on the variance in performance for smaller k, we can not rule out the potential benefit of a more sophisticated selection process and leave it for future work.

7 Conclusion and Future Work

The findings of this paper serve to reinforce the idea that larger language models are not always inherently superior at every task and should therefore not automatically be considered the default choice. We have shown how primitive fasttext models can be competitive with large transformer based language models for syntactically trivial but semantically complex tasks such as cross-lingual metaphor detection of adjective-noun phrases. Furthermore, these small models can easily be enhanced to out-

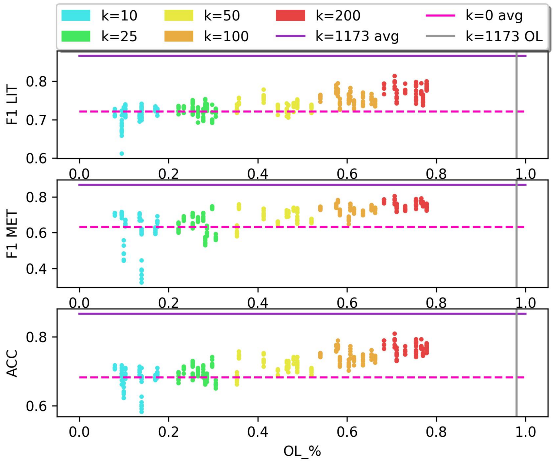


Figure 5: Scatter plot of the different fasttext seeds for rising k and exemplary for EN to PL. We distinguish between the F1 score of literals, metaphors and accuracy. The points are coloured for k and scattered by their percentage of seen adjectives of the test data through the k-shot data (OL%). Line plots are provided for the average performance and adjective overlap when including the whole training data and the average performance for zero-shot.

perform their substantially larger competitors by softening the zero-shot limitation and including small amounts of data from the target language. Based on our experiments, we recommend using k-shot data as a quick and cost-effective measure, over upscaling to a substantially bigger language model. This has the benefit of a computationally less demanding training environment, almost immediate results and a more environmentally friendly model.[10] The claim about static word embeddings being ill-suited for metaphor detection (Maudslay and Teufel (2022)) can not be validated by our findings, since they performed similarly to the contextualized representations. Additionally, we showed that the k-shot paradigm can also benefit architectures utilizing automatically translated training data.

Investigation of the peculiar behaviour of ChatGPT's dependence on the prompts, as seen in Table 4, would be interesting but was beyond the scope of this paper. Finally, we leave expanding the k-shot experiments to sequence labeling for metaphor detection open for future work.

[10]Finetuning the BERT models took more than 3 hours, while our training of our fasttext models concluded after just 30 seconds.

Acknowledgements

We thank Theresa Sick for providing us with the corpus of annotated adjective-noun phrases in German and Pola Stawiany for providing the Polish translation of the Metaphor Identification Procedure and insight into the Polish corpus and translated phrases.

References

Minjin Choi, Sunkyung Lee, Eunseong Choi, Heesoo Park, Junhyuk Lee, Dongwon Lee, and Jongwuk Lee. 2021. MelBERT: Metaphor detection via contextualized late interaction using metaphorical identification theories. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1763–1773, Online. Association for Computational Linguistics.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

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*.

Erik-Lân Do Dinh and Iryna Gurevych. 2016. Token-level metaphor detection using neural networks. In *Proceedings of the Fourth Workshop on Metaphor in NLP*, pages 28–33, San Diego, California. Association for Computational Linguistics.

Steffen Eger, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2018. Cross-lingual argumentation mining: Machine translation (and a bit of projection) is all you need! *arXiv preprint arXiv:1807.08998*.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

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 (LREC 2018)*.

Pragglejaz Group. 2007. Mip: A method for identifying metaphorically used words in discourse. *Metaphor and symbol*, 22(1):1–39.

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

Tsung-Yuan Hsu, Chi-Liang Liu, and Hung-Yi Lee. 2019. Zero-shot reading comprehension by cross-lingual transfer learning with multi-lingual language representation model.

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018a. Loss in translation: Learning bilingual word mapping with a retrieval criterion. *arXiv preprint arXiv:1804.07745*.

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018b. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2984, Brussels, Belgium. Association for Computational Linguistics.

Phillip Keung, Yichao Lu, Julian Salazar, and Vikas Bhardwaj. 2020a. Don't use english dev: On the zero-shot cross-lingual evaluation of contextual embeddings. *arXiv preprint arXiv:2004.15001*.

Phillip Keung, Yichao Lu, Julian Salazar, and Vikas Bhardwaj. 2020b. Don't use English dev: On the zero-shot cross-lingual evaluation of contextual embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 549–554, Online. Association for Computational Linguistics.

Maximilian Köper and Sabine Schulte im Walde. 2017. Improving verb metaphor detection by propagating abstractness to words, phrases and individual senses. In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, pages 24–30.

George Lakoff and Mark Johnson. 1980. *Metaphors we live by*. University of Chicago press.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Xiangji Huang. 2023. A systematic study and comprehensive evaluation of chatgpt on benchmark datasets. *arXiv preprint arXiv:2305.18486*.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

Chee Wee (Ben) Leong, Beata Beigman Klebanov, Chris Hamill, Egon Stemle, Rutuja Ubale, and Xianyang Chen. 2020. A report on the 2020 VUA and TOEFL metaphor detection shared task. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 18–29, Online. Association for Computational Linguistics.

Rowan Hall Maudslay and Simone Teufel. 2022. Metaphorical polysemy detection: Conventional metaphor meets word sense disambiguation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 65–77, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Agnieszka Mykowiecka, Malgorzata Marciniak, and Aleksander Wawer. 2018. Literal, metphorical or both? detecting metaphoricity in isolated adjective-noun phrases. In *Proceedings of the Workshop on Figurative Language Processing*, pages 27–33.

Debora Nozza. 2021. Exposing the limits of zero-shot cross-lingual hate speech detection. 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 907–914, Online. Association for Computational Linguistics.

Adam Przepiórkowski and Agnieszka Patejuk. 2014. Koordynacja leksykalno-semantyczna w systemie współczesnej polszczyzny (na materiale narodowego korpusu języka polskiego). *Język Polski*, pages 104–115.

Elisa Sanchez-Bayona and Rodrigo Agerri. 2022. Leveraging a new Spanish corpus for multilingual and cross-lingual metaphor detection. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 228–240, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Felix Schneider, Sven Sickert, Phillip Brandes, Sophie Marshall, and Joachim Denzler. 2022. Metaphor detection for low resource languages: From zero-shot to few-shot learning in Middle High German. In *Proceedings of the 18th Workshop on Multiword Expressions @LREC2022*, pages 75–80, Marseille, France. European Language Resources Association.

Sabine Schulte im Walde. 2022. Figurative language in noun compound models across target properties, domains and time. In *Proceedings of the 18th Workshop on Multiword Expressions @LREC2022*, page 1, Marseille, France. European Language Resources Association.

Theresa Sick. 2020. Metaphor detection in german adjective-noun-pairs. B.A. thesis, Heidelberg University.

Gerard Steen, Lettie Dorst, Berenike Herrmann, Anna Kaal, Tina Krennmayr, and Trijntje Pasma. 2010. A method for linguistic metaphor identification from mip to mipvu preface. *Method For Linguistic Metaphor Identification: From MIP To MIPVU*, 14:IX–+.

Paul H Thibodeau and Lera Boroditsky. 2011. Metaphors we think with: The role of metaphor in reasoning. *PloS one*, 6(2):e16782.

Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 248–258.

Yulia Tsvetkov, Elena Mukomel, and Anatole Gershman. 2013. Cross-lingual metaphor detection using common semantic features. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 45–51.

Peter Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 680–690, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Yorick Wilks. 1975. A preferential, pattern-seeking, semantics for natural language inference. *Artificial intelligence*, 6(1):53–74.

Yorick Wilks, Adam Dalton, James Allen, and Lucian Galescu. 2013. Automatic metaphor detection using large-scale lexical resources and conventional metaphor extraction. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 36–44, Atlanta, Georgia. Association for Computational Linguistics.

Chenhan Yuan, Qianqian Xie, and Sophia Ananiadou. 2023. Zero-shot temporal relation extraction with ChatGPT. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 92–102, Toronto, Canada. Association for Computational Linguistics.

Appendix

A ChatGPT Prompts

role	content
system	You are a multilingual metaphor detection system. You classify incoming adjective-noun phrases acording to their metaphoricity. Returning 1 if a phrase is a metaphor and returning 0 if a phrase is literal. You return your answers in JSON format, with the prediction at attribute 'label'.
user	[{"id": 1,"phrase": "grünes Landschaftsparadies"}, {"id": 2,"phrase": "knapper Fragenbogen"}, {"id": 3,"phrase": "jährlicher Fleischverzehr"}, {"id": 4,"phrase": "feine Grilladen"}, {"id": 5,"phrase": "amerikanische Erforscher"}. . .]

Table 5: Example prompt for the **ChatGPT** model

role	content
system	You are a multilingual metaphor detection system. You classify incoming adjective-noun phrases acording to their metaphoricity. Returning 1 if a phrase is a metaphor and returning 0 if a phrase is literal. You return your answers in JSON format, with the prediction at attribute 'label'.
user	[{"id": 1,"phrase": "unreadable face"}, {"id": 2,"phrase": "drowsy heat"}, {"id": 3,"phrase": "turbulent water"}, {"id": 4,"phrase": "smokey eyes"}, {"id": 5,"phrase": "metallic surface"}. . .]
assistant	[{"id": 1,"phrase": "unreadable face", "label": 1 }, {"id": 2,"phrase": "drowsy heat", "label": 1 }, {"id": 3,"phrase": "turbulent water", "label": 0 }, {"id": 4,"phrase": "smokey eyes", "label": 1}, {"id": 5,"phrase": "metallic surface", "label": 1 }. . .]
user	[{"id": 1,"phrase": "grünes Landschaftsparadies"}, {"id": 2,"phrase": "knapper Fragenbogen"}, {"id": 3,"phrase": "jährlicher Fleischverzehr"}, {"id": 4,"phrase": "feine Grilladen"}, {"id": 5,"phrase": "amerikanische Erforscher"}. . .]

Table 6: Example prompt for the **ChatGPT+ex** model

role	content
system	You are a multilingual metaphor detection system. You classify incoming adjective-noun phrases according to their metaphoricity based on the Metaphor Identification Procedure. Returning 1 if a phrase is a metaphor and returning 0 if a phrase is literal. You return your answers in JSON format, with the prediction at attribute 'label'. This is the Metaphor Identification Procedure: 1. Read the text to get a general understanding of the meaning 2. Determine the lexical units 3a. Establish the contextual meaning of the unit 3b. Determine if it has a more basic meaning. Basic meaning 'more concrete, body-related, more precise, historically older; not necessarily the most frequent meaning! Does the contextual meaning contrast with the basic meaning but can it be understood in comparison with it? 4. If yes, mark the unit as metaphorical.
user	[{"id": 1,"phrase": "grünes Landschaftsparadies"}, {"id": 2,"phrase": "knapper Fragenbogen"}, {"id": 3,"phrase": "jährlicher Fleischverzehr"}, {"id": 4,"phrase": "feine Grilladen"}, {"id": 5,"phrase": "amerikanische Erforscher"} . . .]

Table 7: Example prompt for the **ChatGPT+MIP** model

Geometry-Aware Supertagging with Heterogeneous Dynamic Convolutions

Konstantinos Kogkalidis[◇,□] and **Michael Moortgat**[□]

[◇]Aalto University
[□]Utrecht Institute of Linguistics OTS, Utrecht University
kokos.kogkalidis@aalto.fi, m.j.moortgat@uu.nl

Abstract

The syntactic categories of categorial grammar formalisms are structured units made of smaller, indivisible primitives, bound together by the underlying grammar's category formation rules. In the trending approach of constructive supertagging, neural models are increasingly made aware of the internal category structure. In turn, this enables them to more reliably predict rare and out-of-vocabulary categories, with significant implications for grammars previously deemed too complex to find practical use. In this work, we revisit constructive supertagging from a graph-theoretic perspective, and propose a framework based on heterogeneous dynamic graph convolutions, aimed at exploiting the distinctive structure of a supertagger's output space. We test our approach on a number of categorial grammar datasets spanning different languages and grammar formalisms, achieving substantial improvements over previous state of the art scores.

1 Introduction

Their close affinity to logics and lambda calculi has made categorial grammars a standard tool of trade for the formally-inclined NLP practitioner. Modern flavors of categorial grammar, despite their (sometimes striking) divergences, share a common architecture. At its core, a categorial grammar is a formal system consisting of two parts. First, there is a *lexicon*, a mapping that assigns to each word a set of *categories*. Categories are quasi-logical formulas recursively built out of atomic categories by means of category forming operations. The inventory of category forming operations at the minimum has the ability to express linguistic function-argument structure. If so desired, the inventory can be extended with extra operations, e.g. to handle syntactic phenomena beyond simple concatenation, or to express additional layers of grammatical information. The second component of the grammar is a small set of *inference rules*, formulated in terms of the category forming operations. The inference rules dictate how categories interact and, through this interaction, how words combine to form larger phrases. Parsing thus becomes a process of deduction comparable (or equatable, depending on the grammar's formal rigor) to program synthesis, providing a clean and elegant syntax-semantics interface.

In the post-neural era, these two components allow differentiable implementations. The fixed lexicon is replaced by *supertagging*, a process that contextually decides on the most appropriate supertags (i.e. categories), whereas the choice of which rules of inference to apply is usually deferred to a parser further down the processing pipeline. The highly lexicalized nature of categorial grammars thus shifts the bulk of the weight of a parse to the supertagging component, as its assignments and their internal make-up inform and guide the parser's decisions.

In this work, we revisit supertagging from a geometric angle. We first note that the supertagger's output space consists of a sequence of trees, which has as of yet found no explicit representational treatment. Capitalizing on this insight, we employ a framework based on heterogeneous dynamic graph convolutions, and show that such an approach can yield substantial improvements in predictive accuracy across categories both frequently and rarely encountered during a supertagger's training phase.

2 Background

The supertagging problem revolves around the design and training of a function tasked with mapping each word (in the context of a sentence) to a category, thus inducing a sequence of categories $\{c_1, \ldots, c_n\}$ from a sentence $\{w_1, \ldots, w_n\}$. Existing supertagging architectures differ in how they implement this mapping, with each implementation choice boiling down to (i) which of the temporal

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 107–119
September 11–12, 2023.

and structural dependencies within and between the input and output are taken into consideration, and (ii) how these dependencies are materialized.

Earlier work would utilize solely occurrence counts from a training corpus to independently map word n-grams to their most likely categories, and then attempt to filter out implausible sequences via rule-constrained probabilistic models (Bangalore and Joshi, 1999). The shift from sparse feature vectors to distributed word representations facilitated integration with neural networks and improved generalization on the mapping domain, extending it to rare and previously unseen words (Lewis and Steedman, 2014). Later, the advent of recurrent neural networks offered a natural means of incorporating temporal structure, widening the input receptive field through contextualized word representations on the one hand (Xu et al., 2015), but also permitting an autoregressive formulation of the output generation, whereby the effect of a category assignment could percolate through the remainder of the output sequence (Vaswani et al., 2016). Regardless of implementation specifics, the discriminative paradigm employed by all above works fails to account for the skewness of the data; exceedingly rare categories are practically impossible to learn, and categories absent from the training data are completely ignored.

As an alternative, the recently emerging constructive paradigm seeks to explore the structure hidden *within* categories. By inspecting their formation rules, Kogkalidis et al. (2019) equates categories to CFG derivations, viewing *each* category as a tiny compositional expression, and a category sequence as the concatenation of their flattened depth-first projections. The goal sequence is now incrementally generated on a symbol-by-symbol basis using a transformer-based seq2seq model; a twist which provides the decoder with the means to construct novel categories on demand, bolstering co-domain generalization. The decoder's global receptive field, however, comes at the heavy price of quadratic memory complexity, which also bodes poorly with the elongated output sequences, leading to a slowed down inference speed. Expanding on the idea, Prange et al. (2021) explicates the categories' tree structure, embedding symbols based on their tree positions and propagating contextualized representations through tree edges, using either residual dense connections or a tree-structured GRU. This adaptation completely eliminates the burden of learning *how* the categorial trees are constructed, instead allowing the model to focus on *what* trees to construct, leading to drastically improved performance. Simultaneously, since the decoder is now token-separable, it permits construction of categories for the entire sentence in parallel, speeding up inference and reducing the network's memory footprint. In the process, however, it loses the ability to model interactions between autoregressed nodes belonging to different trees, morally reducing the task once more to sequence classification (albeit now with a dynamic classifier).

Despite their common goal of accounting for syntactic categories in the zipfian tail, there are tension points between the above two approaches. In flattening categories and concatenating them together, the first breaks the input-to-output alignment and obfuscates the categorial tree structure. In opting for a tree-wise bottom-up decoding, the second forgets about meaningful *inter*-tree output-to-output dependencies. In this paper, we seek to resolve these tension points with a novel, unified and grammar-agnostic supertagging framework based on heterogeneous dynamic graph convolutions. Our architecture combines the merits of explicit tree structures, strong autoregressive properties, near-constant decoding time, and a memory complexity that scales with the input, boasting high performance across the full span of the frequency spectrum and surpassing previously established benchmarks on all datasets considered.

3 Methodology

3.1 Breadth-First Parallel Decoding

Despite seeming at odds, both architectures described fall victim to the same trap of conflating problem-specific structural biases and general purpose decoding orders: one forgets about tree structure in opting for a sequential decoding, whereas the other does the exact opposite, forgetting about sequential structure in opting for a tree-like decoding. We note first that the target output is (a batch of) neither sequences nor trees, but rather *sequences of trees*. Having done that, our task is of a purely technical nature: we simply need to come up with the spatiotemporal dependencies that abide by *both* structural axes, and then a neural architecture that can accommodate them.

Prange et al. (2021) make a compelling case for depth-parallel decoding, given that it's incredibly fast (i.e., not temporally bottlenecked by left-to-right sequential dependencies) but also structurally elegant (trees are only built when/if licensed by non-terminal nodes, ensuring structural correctness virtually for free). Sticking with depth-parallel decoding means necessarily foregoing some autoregressive interactions: we certainly cannot look to the future (i.e., tree nodes located deeper than the current level, since these should depend on the decision we are about to make), but neither to the present (i.e., tree nodes residing in the current level, since these will be all decided simultaneously). This still leaves some leeway as to what could constitute the prediction context. The maximalist position we adopt here is nothing less than the entire past, i.e. *all* the nodes we have so far decoded. Crucially, this extends beyond the ancestry-bound "vertical interactions" of a tree unfolding function implemented *à la* treeRNN, allowing "diagonal" interactions between autoregressed nodes living in different trees.

Such exotic interactions do not follow the inductive biases of any run-of-the-mill architecture, forcing us to turn our attention to structure-aware dynamic convolutions. To make the architecture conducive to learning while keeping its memory footprint in check, we repurpose the encoder's word vectors from initial seeds to recurrent state-tracking vectors that arbitrate the decoding process across both *sequence length* and *tree depth*, respecting the "regularly irregular" structure of the output space. In high level terms, the process can be summarized as an iteration of three alternating stages of message passing rounds.

1. Lexical state vectors are initialized by some external encoder.
2. An empty fringe consisting of blank nodes is instantiated, one such node per word, rooting the corresponding categorial trees.
3. Until a fix-point is reached (there is no longer any fringe):
   (i) **Node Prediction** States project class weights to their respective fringe nodes in a one-to-many fashion. Depending on the arity of the decoded symbols, a next fringe of unfilled nodes is constructed at the appropriate positions; e.g., binary operators expand the fringe by introducing two new blank nodes located directly above them.
   (ii) **Autoregressive Feedback** Each state vector receives feedback in a many-to-one fashion, originating from the lexically-aligned nodes just decoded (i.e., the fringe at the previous time step). This way, state vectors are iteratively updated and progressively aggregate information from the tree as it is being dynamically constructed.
   (iii) **Sequential Feedback** The updated state vectors emit and receive messages to one another in a many-to-many fashion, allowing states to be informed by the decoding progress of their neighbors.

For a visual rendition, refer to Appendix A.

3.2 Architecture

We now move on to detail the individual blocks that together make up the network's pipeline.

3.2.1 Node Embeddings

State vectors are temporally dynamic; they are initially supplied by an external encoder, and are then updated through a repeated sequence of three message passing rounds, described in the next subsections. Tree nodes, on the other hand, are not subject to temporal updates, but instead become dynamically "revealed" by the decoding process. Their representations are computed on the basis of (i) their primitive symbol and (ii) their position within a tree.

Primitive symbol embeddings are obtained from a standard embedding table $W_e : \mathcal{S} \rightarrow \mathbb{R}^{d_n}$ that contains a distinct vector for each symbol in the set of primitives $\mathcal{S}$. When it comes to embedding positions, we are presented with a number of options. It would be straightforward to fix a vocabulary of positions, and learn a distinct vector for each. Such an approach would however lack elegance, as it would impose an ad-hoc bound to the shape of trees that can be encoded (contradicting the constructive paradigm), while also failing to account for the compositional nature of trees. We thus opt for a path-based approach, inspired by and improving upon the idea of Shiv and Quirk (2019). We note first that *paths* over binary branching trees form a semi-group, i.e. they consist of two primitives (namely a left and a right path), and an associative non-commutative binary operator that binds two paths together into a single new one. The archetypical example of a semigroup is matrix multiplication; we therefore instantiate a tensor $P \in \mathbb{R}^{2 \times n_d \times n_d}$ encoding each of the two path primitives as a linear map over symbol embeddings. From the above we can derive a function p that converts positions to linear maps, by performing consecutive matrix multiplications of the primitive weights, as indexed by the binary word of a node's position; e.g. the linear map corresponding to position $12_{10} = 1100_2$ would be $p(12) = P_0P_0P_1P_1 \in \mathbb{R}^{d_n \times d_n}$. We flatten the final map by evaluating it against an initial seed vector ρ_0, corresponding to the tree root.[1] To stabilize training and avoid vanishing or exploding weights, we model paths as *unitary* transformations by parameterizing the two matrices of P to orthogonality using the exponentiation trick on skew-symmetric bases (Bader et al., 2019; Lezcano Casado, 2019).

The representation $n_{i,k}$ of a tree node $\sigma \in \mathcal{S}$ occupying position k in tree i will then be given as the element-wise product of its tree-positional and content embeddings:

$$n_{i,k} = p(k)(\rho_0) \odot (W_e(\sigma)) \in \mathbb{R}^{d_n}$$

The embedder is then essentially an instantiation of a binary branching unitary RNN (Arjovsky et al., 2016), where the choice of which hidden-to-hidden map to follow at each step depends on the node's position relative to its ancestor.[2] Since paths are shared across trees, their representations are in practice efficiently computed once per batch for each unique tree position during training, and stored as fixed embeddings during inference.

3.2.2 Node Prediction

Assuming at step τ a sequence of globally contextualized states h^τ, we need to use each element h_i^τ to obtain class weights for all of the node neighborhood $\mathcal{N}_{i,\tau}$ consisting of all nodes (if any) of tree i that lie at depth τ. We start by down-projecting the state vector into the node's dimensionality using a linear map W_n. The resulting feature vectors are indistinguishable between all nodes of the same tree – to tell them apart (and obtain a unique prediction for each), we gate the feature vectors against each node's positional embedding. From the latter, we obtain class weights by matrix multiplying them against the transpose of the symbol embedding table (Press and Wolf, 2017):

$$\text{weights}_{i,k} = (p(k)(\rho_0) \odot W_n h_i^\tau)\, W_e^\top$$

The above weights are converted into a probability distribution over the alphabet symbols $\mathcal{S}$ by application of the $\mathrm{softmax}$ function.

3.2.3 Autoregressive Feedback

We update states with information from the last decoded nodes using a heterogeneous message-passing scheme based on graph attention networks (Veličković et al., 2018; Brody et al., 2021). First, we use a bottleneck layer W_b to down-project the state vector into the nodes' dimensionality. For each position i and corresponding state h_i^τ, we compute a self-loop score:

$$\tilde{\alpha}_{i,\circlearrowleft,\tau} = w_a \cdot (W_b(h_i^\tau) \mid\mid \mathbf{0})$$

where $w_a \in \mathbb{R}^{2d_n}$ a dot-product weight and $\mathbf{0}$ a d_n-dimensional zero vector. Then we use the (now decoded) neighborhood $\mathcal{N}_{i,\tau}$ to generate a heterogeneous attention score for each node $n_{i,k} \in \mathcal{N}_{i,\tau}$:

$$\tilde{\alpha}_{i,k,\tau} = w_a \cdot (h_i^\tau \mid\mid n_{i,k})$$

Scores are passed through a leaky rectifier non-linearity before being normalized to attention coefficients α. These are used as weighting factors that scale the self-loop and input messages, the latter upscaled by a linear map W_m:

$$\tilde{h}_i^\tau = \sum_{n_{i,k} \in \mathcal{N}_{i,\tau}} \alpha_{i,k,\tau} W_m n_{i,k} + \alpha_{i,\circlearrowleft,\tau} h_i^\tau$$

This can also be seen as a dynamic residual connection – $\alpha_{i,\circlearrowleft,\tau}$ acts as a gate that decides how open the state's representation should be to node feedback (or conversely, how strongly it should retain its current values). States receiving no node feedback (i.e. states that have completed decoding one or more time steps ago) are thus protected from updates, preserving their content. In practice, attention coefficients and message vectors are computed for multiple attention heads independently, but these are omitted from the above equations to avoid cluttering the notation.

3.2.4 Sequential Feedback

At the end of the node feedback stage, we are left with a sequence of locally contextualized states $\tilde{h}_i^\tau$. The sequential structure can be seen as a fully connected directed graph, nodes being states (words) and edges tabulated as the square matrix $\mathcal{E}$, with entry $\mathcal{E}_{i,j}$ containing the relative distance between words i and j. We embed these distances into the encoder's vector space using an embedding table

[1] In practice, paths are efficiently computed once per batch for each unique tree position during training, and stored as fixed embeddings during inference.

[2] Concurrently, Bernardy and Lappin (2022) follow a similar approach in teaching a unitary RNN to recognize Dyck words, and find the unitary representations learned to respect the compositional properties of the task. Here we go the other way around, using the unitary recurrence exactly because we expect them to respect the compositional properties of the task.

$W_r \in \mathbb{R}^{2\kappa \times d_w}$, where κ the maximum allowed distance, a hyper-parameter. Edges escaping the maximum distance threshold are truncated rather than clipped, in order to preserve memory and facilitate training, leading to a natural segmentation of the sentence into (overlapping) chunks. Following standard practices, we project states into query, key and value vectors, and compute the attention scores between words i and j using relative-position weighted attention (Shaw et al., 2018):

$$\tilde{a}_{i,j} = d_w^{-1/2} \left(W_q \tilde{h}_i^{\tau} \odot W_r \mathcal{E}_{i,j}\right) \cdot W_k \tilde{h}_j^{\tau}$$

From the normalized attention scores we obtain a new set of aggregated messages:

$$m'_{i,t} = \sum_{j \in \{0..s\}} \frac{\exp(\tilde{a}_{i,j}) W_v \tilde{h}_j^{\tau}}{\sum_{k \in \{0..s\}} \exp(\tilde{a}_{i,k})}$$

Same as before, queries, keys, values, edge embeddings and attention coefficients are distributed over many heads. Aggregated messages are passed through a swish-gated feed-forward layer (Dauphin et al., 2017; Shazeer, 2020) to yield the next sequence of state vectors:

$$h_i^{\tau+1} = W_3 \left(\text{swish}_1(W_1 m'_{i,\tau}) \odot W_2 m'_{i,\tau}\right)$$

where $W_{1,2}$ are linear maps from the encoder's dimensionality to an intermediate dimensionality, and vice versa for W_3.

3.2.5 Putting Things Together

We compose the previously detailed components into a single layer, which acts a sequence-wide, recurrent-in-depth decoder. We insert skip connections between the input and output of the message-passing and feed-forward layers (He et al., 2016), and subsequently normalize each using root mean square normalization (Zhang and Sennrich, 2019).

4 Experiments

We employ our supertagging architecture in a range of diverse categorial grammar datasets spanning different languages and underlying grammar formalisms. In all our experiments, we bind our model to a monolingual BERT-style language model used as an external encoder, fine-tuned during training (Devlin et al., 2018). In order to homogenize the tokenization between the one directed by each dataset and the one required by the encoder, we make use of a simple localized attention aggregation scheme. The subword tokens together comprising a single word are independently projected to scalar values through a shallow feed-forward layer. Scalar values are softmaxed within their local group to yield attention coefficients over their respective BERT vectors, which are then summed together, in a process reminiscent of a cluster-wide attentive pooling (Li et al., 2016). In cases of data-level tokenization treating multiple words as a single unit (i.e. assigning one type to what BERT perceives as many words), we mark all words following the first with a special `[MWU]` token, signifying they need to be merged to the left. This effectively adds an extra output symbol to the decoder, which is now forced to do double duty as a sequence chunker. To avoid sequence misalignments and metric shifts during evaluation, we follow the merges dictated by the ground truth labels, and consider the decoder's output as correct only if all participating predictions match, assuming no implicit chunking oracles.

4.1 Datasets

We conduct experiments on the two variants of the English CCGBank, the French TLGbank and the Dutch Æthel proofbank. A high-level overview of the datasets is presented in Table 1, and short descriptions are provided in the following paragraphs. We refer the reader to the corresponding literature for a more detailed exposition.

	CCGbank		***TLGbank***	***Æthel***
	original	*rebank*		
Primitives	37	40	27	81
Zeroary	35	38	19	31
Binary	2	2	8	50
Categories	1323	1619	851	5762
in train	1286	1575	803	5146
depth avg.	1.94	1.96	1.99	1.82
depth max.	6	6	7	35
Test Sentences	2407	2407	1571	5770
length avg.	23.00	24.27	27.58	16.52
Test Tokens	55371	56395	44302	95331
Frequent (100+)	54825	55690	43289	91503
Uncommon (10-99)	442	563	833	2639
Rare (1-9)	75	107	149	826
Unseen (OOV)	22	27	31	363

Table 1: Bird's eye view of datasets employed and relevant statistics. Test tokens are binned according to their corresponding categories' occurrence count in the respective dataset's training set. Token counts are measured before pre-processing. Unique primitives for the type-logical datasets are counted after binarization.

CCGBank The English CCGbank (original) (Hockenmaier and Steedman, 2007) and its refined version (rebank) (Honnibal et al., 2010) are resources of Combinatory Categorial Grammar (CCG) derivations obtained from the Penn Treebank (Taylor et al., 2003). CCG (Steedman and Baldridge, 2011) builds lexical categories with the aid of two binary slash operators, capturing forward and backward function application. Some additional rules lent from combinatory logic (Curry et al., 1958) permit constrained forms of type raising and function composition, allowing categories to remain relatively short and uncomplicated while keeping parsing complexity in check. The key difference between the two versions lies in their tokenization and the plurality of categories assigned, the latter containing more assignments and a more fine-grained set of syntactic primitives, which in turn make it a slightly more challenging evaluation benchmark.

French TLGbank The French type-logical treebank (Moot, 2015) is a collection of proofs extracted from the French treebank (Abeillé et al., 2003). The theory underlying the resource is that of Multi-Modal Typelogical Grammars (Moortgat, 1996); annotations are deliberately made compatible with Displacement Calculus (Morrill et al., 2011) and First-Order Linear Logic (Moot and Piazza, 2001) at the cost of a small increase in lexical sparsity. In short, the vocabulary of operators is extended with two modalities that find use in licensing or restricting the applicability of rules related to non-local syntactic phenomena. To adapt their representation to our framework, we cast unary operators into pseudo-binaries by inserting an artificial terminal tree in a fixed slot within them. Due to the absence of predetermined train/dev/test splits, we randomize them with a fixed seed at a 80/10/10 ratio and keep them constant between repetitions.

Æthel Our last experimental test bed is Æthel (Kogkalidis et al., 2020a), a dataset of type-logical proofs for written Dutch sentences, automatically extracted from the Lassy-Small corpus (Noord et al., 2013). Æthel is geared towards *semantic* parsing, which means categories employ linear implication $\multimap$ as their single binary operator. An additional layer of dependency information is realized via unary modalities, now lifted to *classes* of operators distinguishing complement and adjunct roles. The grammar assigns concrete instances of polymorphic coordinator types, as a result containing more and sparser categories (some of which distinctively tall); considering also its larger vocabulary of primitives, it makes for a good stress test for our approach. We experiment with the latest available version of the dataset (version `1.0.0a5` at the time of writing). Same as before, we impose a regular tree structure, this time by merging adjunct (resp. complement) markers with the subsequent (resp. preceding) binary operator, which makes for an unambiguous and invertible representational translation.

4.2 Implementation

We implement our model using PyTorch Geometric (Fey and Lenssen, 2019), which provides a high-level interface to efficient low-level protocols, facilitating fast and pad-free graph manipulations. We share a single hyper-parameter setup across all experiments, obtained after a minimal logarithmic search over sensible initial values. Specifically, we set the node dimensionality d_n to 128 with 4 heterogeneous attention heads and the state dimensionality d_w to 768 with 8 homogeneous attention heads. We train using AdamW (Loshchilov and Hutter, 2018) with a batch size of 16, weight decay of 10^{-2}, and a learning rate of 10^{-4}, scaled by a linear warmup and cosine decay schedule over 25 epochs. During training we provide strict teacher forcing and apply feature and edge dropout at 20% chance. Our loss signal is derived as the label-smoothed negative log-likelihood between the network's prediction and the ground truth label (Müller et al., 2019). We procure pretrained base-sized BERT variants from the transformers library (Wolf et al., 2020): RoBERTa for English (Liu et al., 2019), BERTje for Dutch (de Vries et al., 2019) and CamemBERT for French (Martin et al., 2020), which we fine-tune during training, scaling their learning rate by 10% compared to the decoder.

4.3 Results

We perform model selection on the basis of validation accuracy, and gather the corresponding test scores according to the frequency bins of Table 1. Table 2 presents our results compared to relevant published literature. Evidently, our model surpasses established benchmarks in terms of overall accuracy, matching or surpassing the performance of both traditional supertaggers on common categories and constructive ones on the tail end of the frequency distribution.

We observe that the relative gains appear to scale with respect to the task's complexity. In the original version of the CCGbank, our model is only slightly

model	accuracy (%) overall	frequent	uncommon	rare	unseen
CCG (original)					
Symbol Sequential LSTM /w n-gram oracles (Liu et al., 2021)	95.99	96.40	65.83	8.65[!]	
Cross-View Training (Clark et al., 2018)	96.10	–	–	–	n/a
Recursive Tree Addressing (Prange et al., 2021)	96.09	96.44	68.10	**37.40**	**3.03**
BERT Token Classification (Prange et al., 2021)	96.22	**96.58**	70.29	23.17	n/a
Attentive Convolutions (Tian et al., 2020)	**96.25**	**96.64**	71.04	n/a	n/a
Heterogeneous Dynamic Convolutions (this work)	**96.29**$\pm$0.04	**96.61**$\pm$0.04	**72.06**$\pm$0.72	34.45$\pm$1.58	**4.55**$\pm$2.87
CCG (rebank)					
Symbol Sequential Transformer[†] (Kogkalidis et al., 2019)	90.68	91.10	63.65	34.58	**7.41**
TreeGRU (Prange et al., 2021)	94.62	95.10	64.24	25.55	2.47
Recursive Tree Addressing (Prange et al., 2021)	94.70	95.11	68.86	**36.76**	4.94
Token Classification (Prange et al., 2021)	94.83	95.27	68.68	23.99	n/a
Heterogeneous Dynamic Convolutions (this work)	**95.07**$\pm$0.04	**95.45**$\pm$0.04	**71.40**$\pm$1.15	**37.19**$\pm$1.81	3.70$\pm$0.00
French TLGbank					
ELMo & LSTM Classification (Moot, 2019)	93.20	95.10	75.19	25.85	n/a
BERT Token Classification[‡]	**95.93**	**96.44**	81.39	47.45	n/a
Heterogeneous Dynamic Convolutions (this work)	**95.92**$\pm$0.01	96.40$\pm$0.01	**81.48**$\pm$0.97	**55.37**$\pm$1.00	**7.26**$\pm$2.67
Æthel					
Symbol Sequential Transformer[⋆] (Kogkalidis et al., 2020b)	83.67	84.55	64.70	50.58	**24.55**
BERT Token Classification[‡]	93.52	**94.83**	71.85	38.06	n/a
Heterogeneous Dynamic Convolutions (this work)	**94.08**$\pm$0.02	**95.16**$\pm$0.01	**75.55**$\pm$0.02	**58.15**$\pm$0.01	18.37$\pm$2.73

[!]Accuracy over both bins, with a frequency-truncated training set (authors claim no difference when using the full set).
[†]Numbers from Prange et al. (2021).
[‡]Our replication.
[⋆]Model trained and evaluated on an older dataset version and tree sequences spanning less than 140 nodes in total.

Table 2: Model performance across datasets and compared to recent studies. Numbers are taken from the papers cited unless otherwise noted. For our model, we report averages and standard deviations over 6 runs. Bold face fonts indicate (within standard deviation of) highest performance.

superior to the next best performing model (in turn only marginally superior to the token-based classification baseline), whereas in the rebank version the absolute difference is one order of magnitude wider. The effect is even further pronounced for the harder type-logical datasets, which are characterized by a longer tail, leading to performance comparable to CCGbank's for the French TLGbank (despite it being significantly smaller and sparser), and a 10% absolute performance leap for Æthel (despite its unusually tall and complex types). We attribute this to increased returns from performance in the rare and uncommon bins; there is a synergistic effect between the larger population of these bins pronouncing even minor improvements, and acquisition of rarer categories apparently benefiting from the plurality of their respective bins in a self-regularizing manner. Put simply, learning sparse categories is *easier* and *matters more* for grammars containing many rare categories.

Finally, to investigate the relative impact of each network component, we conduct an ablation study where message passing components are removed from their network in their entirety. Removing the state feedback component collapses the network into a token-wise separable recurrence, akin to a graph-featured RNN without a hidden-to-hidden affine map. Removing the node feedback component turns the network into a Universal Transformer (Dehghani et al., 2018) composed with a dynamically adaptive classification head. Removing both is equatable to a 1-to-many contextualized token classification that is structurally unfolded in depth. Our results, presented in Table 3, verify first a positive contribution from both components, indicating the importance of both information sharing axes. In three out of the four datasets, the relative gains of incorporating state feedback outweigh those of node feedback, and are most pronounced in the case of Æthel, likely due to its positionally agnostic types. With the exception of CCGrebank, relinquishing both kinds of feedback largely underperforms having either one, experimentally affirming their compatibility.

	-sf	-nf	-sf-nf
***CCG** (original)*	-0.05	-0.01	-0.08
***CCG** (rebank)*	-0.12	-0.04	-0.07
French TLGbank	-0.13	-0.14	-0.23
Æthel	-0.24	-0.12	-0.37

Table 3: Absolute difference in overall accuracy when removing the state and node feedback components (averages of 3 repetitions).

5 Related Work

Our work bears semblance and owes credit to various contemporary lines of work. From the architectural angle, we perceive our work as an application-specific offspring of weight-tied architectures, dynamic graph convolutions and structure-aware self-attention networks. The depth recurrence of our decoder is inspired by weight-tied architectures (Dehghani et al., 2018; Bai et al., 2019) and their graph-oriented variants (Li et al., 2016), which model neural computation as the fix-point iteration of a single layer against a structured input, thus allowing for a dynamically adaptive computation "depth" – albeit with a constant parameter count. Analogously to structure-aware self-attention networks (Zhu et al., 2019; Cai and Lam, 2020) and graph attentive networks (Veličković et al., 2018; Yun et al., 2019; Ying et al., 2021; Brody et al., 2021), our decoder employs standard query/key and fully-connected attention mechanisms injected with structurally biased representations, either at the edge or at the node level. Finally, akin to dynamic graph approaches (Liao et al., 2019; Pareja et al., 2020), our decoder forms a closed loop system that autoregressively generates its own input, in the process becoming exposed to subgraph structures that drastically differ between time steps.

From the application angle, our proposal is a refinement of and a continuation to recent advances in categorial grammar supertagging. Similar to the transition from words to subword units (Sennrich et al., 2016), constructive supertaggers seek to bolster generalization by disassembling syntactic categories into smaller indivisible units, thereby incorporating structure at a finer granularity scale. The original approach of Kogkalidis et al. (2019) employed seq2seq models to directly translate an input text to a flattened projection of a categorial sequence, demonstrating that the correct prediction of categories unseen during training is indeed feasible. Prange et al. (2021) improved upon the process through the explicit accounting of the tree structure embedded within categorial types, while Liu et al. (2021) explored the orthogonal approach of employing a transition-based "parser" over individual categories. Outside the constructive paradigm, Tian et al. (2020) employed graph convolutions over sentential edges built from static, lexicon-based preferences. Our approach is a bridge between prior works; our modeling choice of structure-aware graph convolutions boasts the merits of ex+plicit sentential and tree-structured edges, a structurally constrained, valid-by-construction output space, favorable memory and time complexities, partial autoregressive context flows, end-to-end differentiability with no vocabulary requirements, and minimal rule-based structure manipulation.

6 Conclusion

We have proposed a novel supertagging methodology, where both the linear order of the output sequence and the tree-like structure of its elements is made explicit. To represent the different information sources (sentential word order, subword contextualized vectors, tree-sequence order and intra-tree edges) and their disparate sizes and scales, we turned to heterogeneous graph attention networks. To capture the autoregressive dependencies between different trees, we formulated the task as a dynamic graph completion process, aligning each subsequent temporal step with a higher order tree node neighborhood and predicting them in parallel across the entire sequence. We tested our methodology on four different datasets spanning three languages and as many grammar formalisms, establishing new state of the art scores in the process. Through our ablation studies, we showed the importance of incorporating both *intra-* and *inter*-tree context flows, to which we attribute our system's performance.

Other than architectural adjustment and optimizations, several interesting ideas present themselves as promising research avenues. First, it is worthwhile to consider adaptations of our framework to either allow an efficient integration of more "exotic" context pathways, e.g. sibling node interactions, or alter the graph's decoding order altogether. On a related note, for formalisms faithful to the linear logic roots of categorial grammars, it seems reasonable to anticipate that the goal graph can be compactified by collapsing primitive nodes of opposite polarity according to their interactions, unifying the tasks of supertagging and parsing with a single end-to-end framework.

Practice aside, our results pose further evidence that lexical sparsity, historically deemed the categorial grammar's curse, might well just require a change of perspective to tame and deploy as the answer to the very problem it poses.

Limitations

Despite its objective success, our methodology is not without limitations. Most importantly, our model trades inference speed for an incompatibility with local greedy algorithms like beam search. Put plainly, obtaining more than the "best" category assignment per word is not straightforward, which can potentially negatively impact the downstream parser's coverage. A possible solution would involve branching across multiple tree-slices (i.e. sequences of partial assignments) rather than single predictions, but efficiently computing scores and comparing between complex structures is uncharted territory and not trivial to implement. Note, however, that the issue is not unique to our system but common to all decoders that perform multiple assignments concurrently.

Parallel or not, all autoregressive decoders assume an order on their output: the standard left-to-right order (which makes sense for text) has become the de facto choice for most applications. The order we have chosen to employ here is structurally faithful to our output, but is neither the only one, nor necessarily the most natural one. In that sense, the entanglement between structural bias (i.e. from the graph operations and representations) and decoding priority (i.e. the order in which trees become revealed) is a practical decision rather than a deep one – a better operationalization could for instance employ an insertion-style operation on the graph-structured output to yield an "easy-first" geometric tagger. We await further developments and community insights on that front.

Finally, the system carries the standard risks of any NLP architecture reliant on machine learning, namely linguistic biases inherited from the unsupervised pretraining of the incorporated language models, and annotation biases derived from the supervised training over human-labeled data.

References

Anne Abeillé, Lionel Clément, and François Toussenel. 2003. Building a treebank for French. In *Treebanks*, pages 165–187. Springer.

Martin Arjovsky, Amar Shah, and Yoshua Bengio. 2016. Unitary evolution recurrent neural networks. In *International conference on machine learning*, pages 1120–1128. PMLR.

Philipp Bader, Sergio Blanes, and Fernando Casas. 2019. Computing the matrix exponential with an optimized Taylor polynomial approximation. *Mathematics*, 7(12):1174.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2019. Deep equilibrium models. *Advances in Neural Information Processing Systems*, 32.

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):237–265.

Jean-Philippe Bernardy and Shalom Lappin. 2022. Assessing the unitary rnn as an end-to-end compositional model of syntax. *arXiv preprint arXiv:2208.05719*.

Shaked Brody, Uri Alon, and Eran Yahav. 2021. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*.

Deng Cai and Wai Lam. 2020. Graph transformer for graph-to-sequence learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7464–7471.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925, Brussels, Belgium. Association for Computational Linguistics.

Haskell Brooks Curry, Robert Feys, William Craig, J Roger Hindley, and Jonathan P Seldin. 1958. *Combinatory Logic*, volume 1. North-Holland Amsterdam.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 933–941.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT model. *arXiv preprint arXiv:1912.09582*.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. 2018. Universal transformers. In *International Conference on Learning Representations*.

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*.

Matthias Fey and Jan E. Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

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.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Matthew Honnibal, James R Curran, and Johan Bos. 2010. Rebanking CCGbank for improved np interpretation. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 207–215.

Konstantinos Kogkalidis, Michael Moortgat, and Tejaswini Deoskar. 2019. Constructive type-logical supertagging with self-attention networks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 113–123, Florence, Italy. Association for Computational Linguistics.

Konstantinos Kogkalidis, Michael Moortgat, and Richard Moot. 2020a. ÆTHEL: Automatically extracted typelogical derivations for Dutch. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5257–5266, Marseille, France. European Language Resources Association.

Konstantinos Kogkalidis, Michael Moortgat, and Richard Moot. 2020b. Neural proof nets. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 26–40, Online. Association for Computational Linguistics.

Mike Lewis and Mark Steedman. 2014. Improved CCG parsing with semi-supervised supertagging. *Transactions of the Association for Computational Linguistics*, 2:327–338.

Mario Lezcano Casado. 2019. Trivializations for gradient-based optimization on manifolds. *Advances in Neural Information Processing Systems*, 32.

Yujia Li, Richard Zemel, Marc Brockschmidt, and Daniel Tarlow. 2016. Gated graph sequence neural networks. In *Proceedings of ICLR'16*.

Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Will Hamilton, David K Duvenaud, Raquel Urtasun, and Richard Zemel. 2019. Efficient graph generation with graph recurrent attention networks. *Advances in Neural Information Processing Systems*, 32.

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*.

Yufang Liu, Tao Ji, Yuanbin Wu, and Man Lan. 2021. Generating CCG categories. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13443–13451.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Michael Moortgat. 1996. Multimodal linguistic inference. *JoLLI*, 5(3/4):349–385.

Richard Moot. 2015. A type-logical treebank for French. *Journal of Language Modelling Vol*, 3(1):229–264.

Richard Moot. 2019. Reconciling vectors with proofs for natural language processing. Compositionality in formal and distributional models of natural language semantics, 26th Workshop on Logic, Language, Information and Computation (WoLLIC 2019). Retrieved from `https://richardmoot.github.io/Slides/WoLLIC2019.pdf`.

Richard Moot and Mario Piazza. 2001. Linguistic applications of first order intuitionistic linear logic. *Journal of Logic, Language and Information*, 10(2):211–232.

Glyn Morrill, Oriol Valentín, and Mario Fadda. 2011. The displacement calculus. *Journal of Logic, Language and Information*, 20(1):1–48.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? *Advances in neural information processing systems*, 32.

Gertjan van Noord, Gosse Bouma, Frank Van Eynde, Daniël de Kok, Jelmer van der Linde, Ineke Schuurman, Erik Tjong Kim Sang, and Vincent Vandeghinste. 2013. Large scale syntactic annotation of written Dutch: Lassy. In *Essential Speech and Language Technology for Dutch*, pages 147–164. Springer.

Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. 2020. EvolveGCN: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5363–5370.

Jakob Prange, Nathan Schneider, and Vivek Srikumar. 2021. Supertagging the long tail with tree-structured decoding of complex categories. *Transactions of the Association for Computational Linguistics*, 9:243–260.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. 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.

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.

Noam Shazeer. 2020. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Vighnesh Shiv and Chris Quirk. 2019. Novel positional encodings to enable tree-based transformers. *Advances in Neural Information Processing Systems*, 32.

Mark Steedman and Jason Baldridge. 2011. Combinatory categorial grammar. In Robert Borsley and Kersti Börjars, editors, *Non-Transformational Syntax: Formal and Explicit Models of Grammar*, pages 181–224. Wiley-Blackwell.

Ann Taylor, Mitchell Marcus, and Beatrice Santorini. 2003. The Penn treebank: an overview. *Treebanks*, pages 5–22.

Yuanhe Tian, Yan Song, and Fei Xia. 2020. Supertagging Combinatory Categorial Grammar with attentive graph convolutional networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6037–6044, Online. Association for Computational Linguistics.

Ashish Vaswani, Yonatan Bisk, Kenji Sagae, and Ryan Musa. 2016. Supertagging with LSTMs. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 232–237, San Diego, California. Association for Computational Linguistics.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

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.

Wenduan Xu, Michael Auli, and Stephen Clark. 2015. CCG supertagging with a recurrent neural network. 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 250–255, Beijing, China. Association for Computational Linguistics.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. 2019. Graph transformer networks. *Advances in neural information processing systems*, 32.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32.

Jie Zhu, Junhui Li, Muhua Zhu, Longhua Qian, Min Zhang, and Guodong Zhou. 2019. Modeling graph structure in transformer for better AMR-to-text generation. 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 5459–5468, Hong Kong, China. Association for Computational Linguistics.

A Visualization of the decoding process

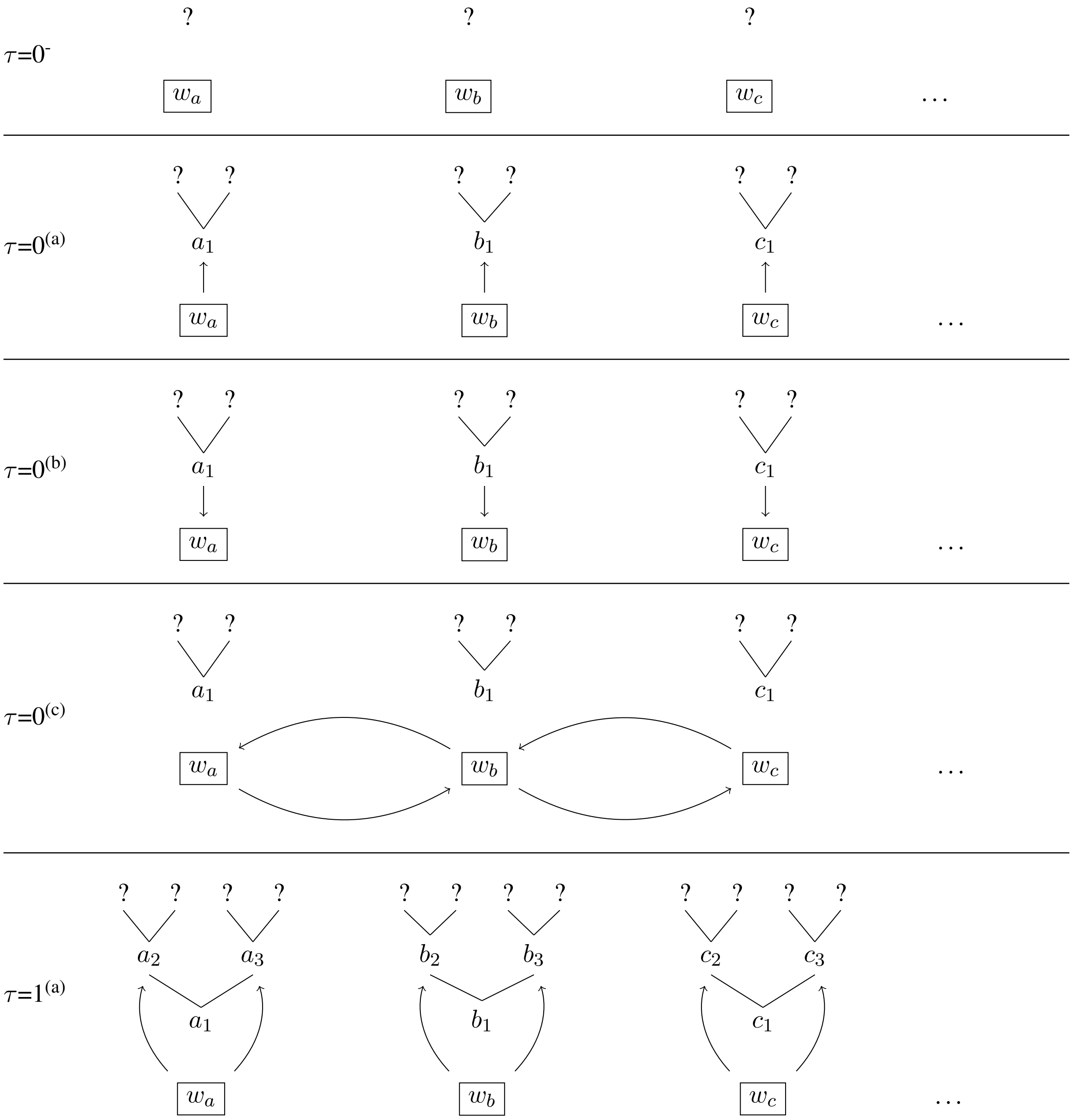
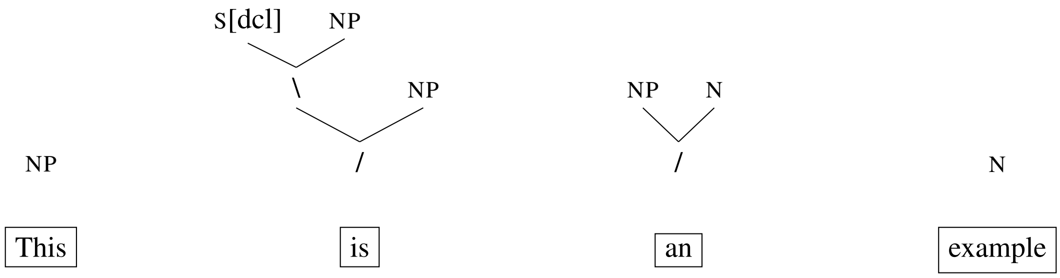
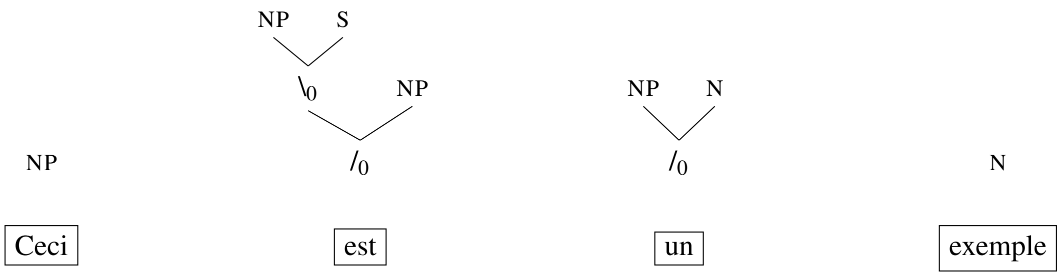


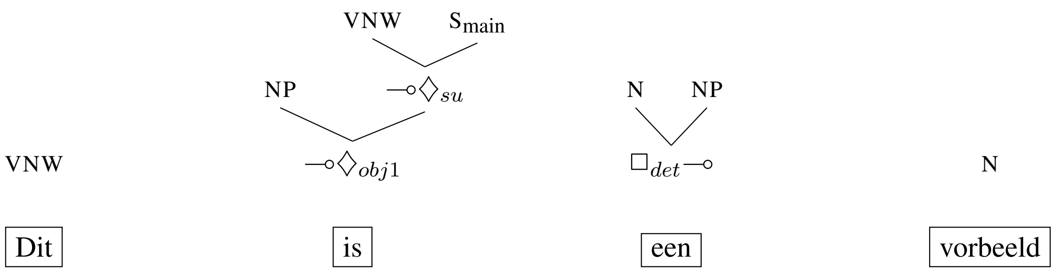
Figure 1: A frame by frame view of the first decoding step, where the abstract canvas assumes words w_a, w_b, w_c $\ldots$, rooting fully binary trees a, b, $c \ldots$, with nodes enumerated in a breadth-first fashion. For an intuition on what a concrete canvas might look like, refer to Figure 2.



(a) CCGbank



(b) French TLGbank



(c) Æthel

Figure 2: Artificial but concrete canvas examples for the three grammars experimented on.

UseClean: learning from complex noisy labels in named entity recognition

Jinjin Tian and **Kun Zhou** and **Meiguo Wang** and **Yu Zhang** and **Benjamin Yao** and **Xiaohu Liu** and **Chenlei Guo**
Alexa AI, Amazon

Abstract

We investigate and refine denoising methods for NER task on data that potentially contains extremely noisy labels from multi-sources. In this paper, we first summarized all possible noise types and noise generation schemes, based on which we built a thorough evaluation system. We then pinpoint the bottleneck of current state-of-art denoising methods using our evaluation system. Correspondingly, we propose several refinements, including using a two-stage framework to avoid error accumulation; a novel confidence score utilizing minimal clean supervision to increase predictive power; an automatic cutoff fitting to save extensive hyper-parameter tuning; a warm started weighted partial CRF to better learn on the noisy tokens. Additionally, we propose to use adaptive sampling to further boost the performance in long-tailed entity settings. Our method improves F1 score by on average at least $5 \sim 10\%$ over current state-of-art across extensive experiments.

1 Introduction

Named Entity Recognition (NER) aims to recognize mentions of rigid designators from text belonging to predefined semantic types such as a person, location, organization, etc. NER not only acts as a standalone tool for information extraction (IE), but also plays an essential role in a variety of natural language processing (NLP) applications such as text understanding, information retrieval, automatic text summarization, question answering, machine translation, and knowledge base construction, etc. Recent progress in deep learning has significantly advanced NER performances (e.g. (Huang et al., 2015; Lample et al., 2016; Li et al., 2020a)). However, in the presence of noisy labels, training DNNs is known to be vulnerable to noisy labels because the significant number of model parameters allow DNNs easily overfit to even corrupted labels. This problem first raised attention in computer vision (CV): (Zhang et al., 2021a) demonstrated that DNNs can easily fit an entire training dataset with any ratio of corrupted labels, which eventually resulted in poor generalizability on a test dataset. Unfortunately, popular regularization techniques, such as data augmentation, weight decay, dropout, and batch normalization do not completely overcome the overfitting issue caused by noisy labels.

Many endeavors have been put into handling noisy labels. Note that this is a fundamentally different problem than general feature-level noise (Zhang and Zhou, 2023; Zheng et al., 2021; Chen et al., 2023; Wang et al., 2021). Except for the specific techniques in certain science domains (Feng et al., 2023), most of those methods are first designed for computer vision or instance-level classification tasks in NLP like text classification. Denoising methods in the NER domain are generally under-explored and rendered harder: for NER, only correct detection of both the entity boundary and entity class are rendered as one correct prediction. Therefore, the label noise in NER is more complex than those in CV or text classification. For example, human annotators could produce mis-specified entity boundaries; other automatic labels generation like distant supervision (Liang et al., 2020) from the dictionary or database often generate incomplete annotations, meaning some entity words are wrongly named as non-entity simply because they are not recorded in the database; others like transfer learning or domain adaptation (Lee et al., 2017; Raghuram et al., 2022; Li and Metsis, 2022) from one domain to another domain could cause wrongly labeled classes for many entity words, as same words could have different semantic types in different domains.

Due to the lack of clean data resources, the majority of denoising literature is unwilling to use any clean validation data or anchor points for denoising, regardless of the fact that most of them often require massive computation cost or extensive hyper-parameter tuning (Song et al., 2022).

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 120–130
September 11–12, 2023.

In fact, those supervision-free methods often suffer from error propagation as the error incurred by false correction/filtering will be accumulated due to lack of supervision, especially when the number of classes or the number of mislabeled examples is large (Shu et al., 2019). To overcome those obstacles, maintaining multiple DNNs or training a DNN in multiple rounds is frequently used (e.g. (Wang et al., 2019; Northcutt et al., 2021)), but these approaches significantly degrade the efficiency of the learning pipeline (Song et al., 2022). In industry-level applications, meta gold datasets (i.e. high-quality/clean datasets) are commonly available, to guarantee direct and reliable evaluation of methods and therefore stable and supreme user experience; also the amount of available data rapidly increases in big companies. More attention should be paid to how to best design and leverage the meta gold dataset to do efficient, effective, and stable label denoising.

Motivated by the above, in this paper, we study the label denoising problem in NER, and we contribute in the following three aspects:

- We build a thorough evaluation system via summarizing all possible noise types and noise generation schemes in NER domain, which was before lacked in the domain. Through this system we find out that the baseline methods [1] are already agnostic to some noise types; while for the other, the noise rate influences the effectiveness of denoising rather than noise type.

- We find out that the current state-of-art denoising method is only effective in very limited noise cases, and the time expensive self-training is often unnecessary due to error propagation. Through careful ablation study, we pinpoint that the true bottleneck of its effectiveness is reliable sample selection.

- We propose an effective and efficient method that leverages minimal clean data to do sample selection and apply weighted semi-supervised learning with a warm start. Under our designed fair comparison [2], our method stably outperforms other state-of-the-art methods across the broad types of simulated noises by a large margin, as well as on realistic data augmentation generated noise. We provide guidelines on further boosting the performance of our method in different application scenarios.

In the following, we will introduce the related work in more detail in Section 2 and provide a formal problem and method description in Section 3. We describe our experiment setting and corresponding results in Section 4, where we also provide a careful ablation study to narrow down the bottleneck of the current state-of-arts method. In the end, we summarize our findings and contributions and some possible future directions in Section 5.

2 Related work

Learning on noisy labels Most of the denoising methods are designed for computer vision (Song et al., 2022), that is, instance level classification. They can generally be categorized into the following four categories: 1) noise modeling: adding a noise adaptation layer at the top of an underlying DNN to learn the transition between clean and noisy labels, e.g.(Chen and Gupta, 2015; Sukhbaatar et al., 2015; Goldberger and Ben-Reuven, 2017)); 2) regularization: enforcing a DNN to overfit less to false-labeled examples explicitly or implicitly, e.g. (Pereyra et al., 2017; Zhang et al., 2018; Menon et al., 2020; Xia et al., 2021; Wei et al., 2021); 3) sample reweighting: adjusting the loss value according to the trust-level of a given sample, e.g. (Wang et al., 2017; Chang et al., 2017; Zhang et al., 2021b; Shu et al., 2019); 4) sample selection: identifying true-labeled examples from noisy training data via multi-network or multi-round learning, e.g. (Han et al., 2018; Jiang et al., 2018; Yu et al., 2019; Wang et al., 2018; Li et al., 2020b; Zhou et al., 2020; Berthelot et al., 2019). From prior work and our investigation, we generally note that, noise modeling type of methods often estimate the transition matrix with large error when only noisy training data is used or when the noise rate is high; regularization type of methods often introduce sensitive model-dependent hyperparameters and therefore hard to stably work in practice; sample reweighting is often more useful for instance level classification, which is not the case in NER problem domain where often a graphical model is adopted for classification; sample selection is well motivated and works well in general, also its has more interpretability and light-weights.

[1]We call methods designed for clean NER data set as baseline methods, and particularly, we choose bert-CFR as our main baseline model of consideration due to its SOTA performance (Lample et al., 2016).

[2]We include the minimal clean data into the training set for all the methods for a fair comparison.

For industry-level application considerations: we hope to seek solutions that are more lightweight, stable, and easy to tune. Therefore we focus on the line of methods using sample selection.

Semi-supervised learning for NER task An inherent limitation of sample selection is to discard all the un-selected training examples, thus resulting in a partial exploration of training data. To exploit all the noisy examples, researchers have attempted to combine sample selection with other orthogonal ideas. The most prominent method in this direction is combining a specific sample selection strategy with a specific semi-supervised learning model (He et al., 2023; Dong et al., 2021). For example, the most promising method in this direction is combining a specific sample selection strategy with a specific semi-supervised learning model like Partial CRF (Tsuboi et al., 2008).

3 Method: UseClean

Our method UseClean is built upon a well-known NER modeling called Conditional Random Field (CRF). Specifically, consider a sentence of words $\boldsymbol{u} : [u_1, \ldots, u_s]$, and a corresponding sequence of tags $\boldsymbol{y} : [y_1, \ldots, y_s]$, where $y_i \in \mathcal{E} := \{1, \ldots, K\}$, CRF (Lample et al., 2016) models the conditional probability of $\boldsymbol{y}$ given $\boldsymbol{u}$ as:

$$p(\boldsymbol{y}|\boldsymbol{u}) \propto \sum_{1 \le i \le s} (T_{y_{i-1}, y_i} + A_{i, y_i}) \in \mathbb{R} \tag{1}$$

$$\text{where } \boldsymbol{A} = Linear(\boldsymbol{h}) \in \mathbb{R}^{s \times K}; \tag{2}$$

$$\boldsymbol{h} = Encoder(\boldsymbol{u}) \in \mathbb{R}^{s \times m}; \tag{3}$$

$$\boldsymbol{T} \in \mathbb{R}^{K \times K}. \tag{4}$$

Here $\boldsymbol{h}$ denotes the encoder hidden representation, $Linear(\cdot)$ denotes a linear layer that converts $\boldsymbol{h}$ into the network estimation for the possibility of y_i at word i given utterance $\boldsymbol{u}$; and the transition score T_{ij} to model the transition from i-th label to j-th for a pair of consecutive time steps, and it is position independent. Dynamic programming can be used efficiently to compute T and inference optimal tag sequences (Sutton et al., 2012).

In the following, we will introduce our two-stage method UseClean built upon this encoder-CRF model. Figure 1 shows the whole working flow of our UseClean method.

3.1 Clean anchor: a better confidence score

NLNCE (Liu et al., 2021) uses the so-called memorization effect observed in computer vision (Arpit et al., 2017; Zhang et al., 2021a). It observes that neural networks usually take precedence over noisy data to fit clean data, which indicates that noisy data are more likely to have larger loss values in the early training epochs. However, we observe that this is not generally true (see the Figure 2 for examples), which in turn leads to many wrong selection and also error accumulation.

Therefore, we propose to use a two-stage framework that uses a little clean supervision to reduce wrong selection and also error accumulation. Specifically, given all the training data, we sample a small portion (around 1-3%) and annotate it with clean labels, then we train a BERT-CRF model on this small gold data. We call this model the clean anchor model.

Then we apply the clean anchor model on the rest of the training data and compute two choices of confidence scores for i-th token in utterance $\boldsymbol{u}$. The marginal probability based score called `Map` from (Liu et al., 2021):

$$r_i = p_{\text{anchor}}(y_i|\boldsymbol{u}) = \alpha_i \beta_i / Z, \tag{5}$$

which measures how likely the i-th token is labeled y_i under the clean anchor model, where β is the backward variable and can be computed with the Backward algorithm; and the logit value differences based score `Diff`:

$$d_i = \max_{j \in [K]} \{A_{i,j}\} - A_{i,y_i}, \tag{6}$$

which measures the gap between the logit of the observed label and the predicted label. We observe no universal winner of those two scores in our extensive experiments, therefore we report the best over them.

Adaptive Sampling. Under the existence of the class imbalance [3], it is very likely that our random sampled small clean dataset does not contain certain tail entity types, and therefore leading to bad separation of clean and noisy tokens in them. To mitigate this effect, we consider a constrained sampling method that tries to sample more from the tail entities: sampling only from utterance that contains at least one tail entity (we define the entities that constitute the tail 20% quantile as the tail entities). In this paper, if a dataset appears to have long

[3]Other popular methods for combating imbalance issue includes the logit adjustment method (Menon et al., 2021), but we did not find it was able to improve the downstream NER performance in our setting.

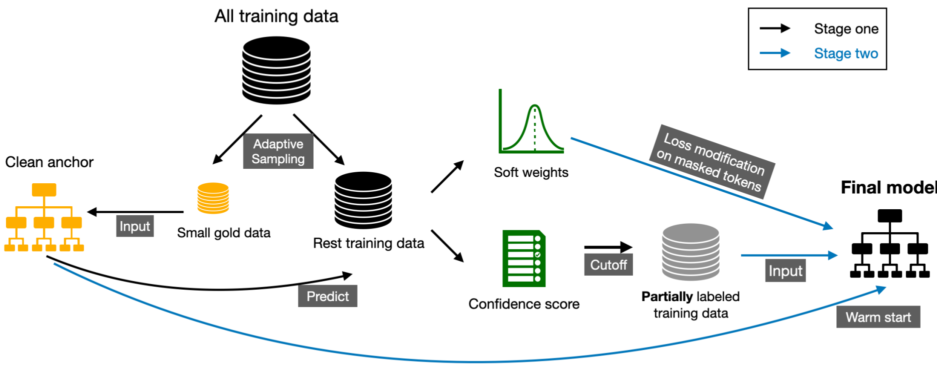


Figure 1: A demonstration of the working flow of UseClean model.

tail entity distribution[4], we will adopt an adaptive sampling scheme: consider both random sampling and constrained sampling and report the best over them.

3.2 FitMix: automatic sample selection

From Figure 2 we can see that, the confidence score of clean and noisy seems to follow a Gamma-Gaussian mixture distribution, where the noisy component follows the Gaussian distribution and the clean component follows the Gamma distribution. So we propose to model the confidence score s as the following:

$$s \sim wf + (1-w)g, \quad \text{where } w \in [0,1] \tag{7}$$

$$f \sim \Gamma(\alpha, \beta),\ g \sim N(\mu, \sigma),\ , \tag{8}$$

and fit all the parameters $(w, \mu, \sigma, \alpha, \beta)$ using Expectation-Maximization algorithm. Then with the fitted parameters $(\widehat{w}, \widehat{\mu}, \widehat{\sigma}, \widehat{\alpha}, \widehat{\beta})$, we can compute the theoretical F1 given a cutoff C in closed form:

$$F_1(C) = \frac{\widehat{w}\Big(1-\Gamma_{\widehat{\alpha},\widehat{\beta}}(C)\Big) + (1-\widehat{w})\Big(2-\Phi_{\widehat{\mu},\widehat{\sigma}}(C)\Big)}{(1-\widehat{w})\Big(1-\Phi_{\widehat{\mu},\widehat{\sigma}}(C)\Big)}.$$

We select C such that $F_1(C)$ is maximized and treat all tokens that have $s > C$ as non-trustworthy.

3.3 Warm weight: learning on noisy tokens

After we do the sample selection, we treat all non-trustworthy tokens as unlabeled and use the idea of semi-supervised learning. Liu et al. (2021) simply sum over all the token sequences that are compatible with the trusted annotations. Specifically, denoting the trusted annotation sequence as $\boldsymbol{y}_p$, from it we can derive a set of all possible complete label sequences that are compatible with the incomplete label sequence, and let us call this set $\mathcal{C}(\boldsymbol{y}_p)$, then semi-supervised loss function can be written as

$$L(\boldsymbol{\theta}) = -\log \sum_{\widetilde{\boldsymbol{y}} \in \mathcal{C}(\boldsymbol{y}_p)} p_\theta(\widetilde{\boldsymbol{y}}|\boldsymbol{u}) \tag{9}$$

Inspired by Jie et al. (2019) for better modeling of NER with incomplete annotations, we instead use a weighted version:

$$L_{weight}(\boldsymbol{\theta}) = -\log \sum_{\widetilde{\boldsymbol{y}} \in \mathcal{C}(\boldsymbol{y}_p)} q_{\mathcal{D}}(\widetilde{\boldsymbol{y}}|\boldsymbol{u}) p_\theta(\widetilde{\boldsymbol{y}}|\boldsymbol{u}), \tag{10}$$

where $q_{\mathcal{D}}$ represents the true data distribution. We estimate $q_{\mathcal{D}}$ as q_{anchor}, which is the distribution computed using our trained clean anchor model. As the clean anchor model is trained on clean data, therefore we believe these weights represent some level of prior information of the underlying true label sequence distribution. By putting more probability mass on a path that is close to the true path, we can guide the model to quickly learn the essential parameters that can correctly predict the true path in the inference stage.

4 Experiments

4.1 Datasets

We consider three datasets for evaluation throughout this paper: an Alexa dialog dataset called Massive (FitzGerald et al., 2022), which contains around 16K samples, and 55 entity types across 18 domains; a popular benchmark dataset CoNLL03 (Sang and De Meulder, 2003), which contains around 20K samples, 4 entity types in News domain; and a Wikipedia dataset Wikigold (Balasuriya et al., 2009), which contains around 1.8K samples over 4 entity types in Wikipedia domain.

[4] Long tail distribution means having many classes of small sizes.

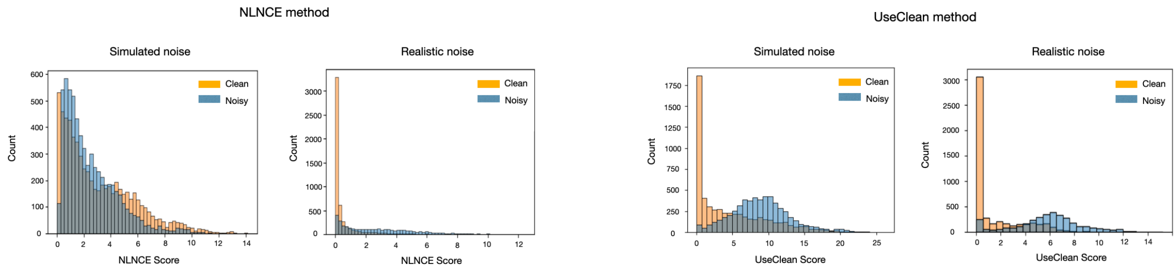


Figure 2: The distribution of confidence score for both simulated noise ("bias" type, detailed description in Section 4) and realistic transfer learning generated noise (detailed description in Section 4), using NLNCE method and our UseClean method. For NLNCE, we plot the distribution of confidence score at epoch 2, we can see that the clean and noisy samples are highly overlapped, i.e. the "early learning" phenomena does not hold true. On the other hand, our UseClean method has better separation of clean and noisy.

Synthetic noisy datesets Given a dataset with labels of good quality, we can treat its original labels as truth and manually perturb it to generate noise. In this paper, we consider randomly selecting $x\%$ of utterance, and random select $\max\{1, 0.2\#\text{entities}\}$ entities (if there are any), and perturb their labels by the different noise generation schemes showed in Table 1. We mainly focus on the high noise rate regime: i.e. $70\%, 100\%$ utterance level noise. We point out that even with the same utterance level noise rate, the word level noise rate can vary a lot for different noise types. For example, `shift` and `shrink` noise types often have much lower word level noise rates compared with others, this is due to the fact that `shift` and `shrink` can only happen on entities with multiple words, while the others can happen on any entity. Due to the size imbalance between entity and nonentity words, we compute the word-level noise rate for entity and nonentity separately. In the following, we use the summation of the entity and nonentity word-level noise rate as the total word-level noise rate for simplicity.

Realistic noise We also consider more realistic noisy label generation. In practice, many cheap labels are generated either from distant supervision or transfer learning.

- *Distant supervision:* We consider three datasets including Massive (FitzGerald et al., 2022), CoNLL03 (Sang and De Meulder, 2003), Wikigold (Balasuriya et al., 2009). In this setting, the distantly supervised tags for CoNLL03 and Wikigold are generated by the dictionary following BOND (Liang et al., 2020), while for massive, we provide distant supervision simply using our own defined dictionary.
- *Transfer learning:* We consider two datasets: Massive (FitzGerald et al., 2022) and CoNLL03 (Sang and De Meulder, 2003). For CoNLL03, we consider transferring the Wikigold dataset to it, as they share exactly the same entity types. To do the transfer, we directly learn a model on Wikigold and predict it on CoNLL03. For Massive, we use data in 9 domains of massive data and transfer them to the rest 9 domains. To make the most meaningful transfer, we compute this domain by domain entity types overlapping matrix, where each cell indicates how many entities types a pair of two domains share. Then we intentionally split the domains into source and target such that the domain pairs with high overlaps are separated, and hence model learned on source domains can have more knowledge transferable to the target domains.

4.2 Methods for comparison

For **Baseline**, we follow the implementation of the neural-CRF model proposed in (Lample et al., 2016) without any denoising steps. Particularly, it models the tag sequence as a linear-chain conditional random field, where only subsequent tags have an edge. Also, we consider the following three NER denoising methods on top of the baseline, which we believe are the most competitive methods in the literature. **CoReg** (Zhou and Chen, 2021) propose a regularization based NER denoising method called CoReg, where the regularization term is based on model agreement. **NLNCE** (Liu et al., 2021) utilize the early learning phenomena and select the noisy tokens via gradually truncating

	noise type	explanation	example: "show me the meetings held last month"
truth	-	-	[O, O, O, S-event_name, O, B-date, I-date]
over/in-complete	miss	label an entity word as nonentity	[O, O, O, S-event_name, O, O, B-date]
	over	label a nonentity word as some random entity	[O, S-person, O, S-event_name, O, B-date, I-date]
boundary error	shift	for an entity contains multiple words, shift its boundary to the left or right by one word.	[O, O, O, S-event_name, B-date, I-date, O]
	extend	for an entity, extend its boundary to the left or right by one word.	[O, O, O, B-event_name, I-even_name, B-date, I-date]
	shrink	for an entity contains multiple words, shrink its boundary from the left or right by one word.	[O, O, O, S-event_name, O, S-date, O]
class error	swap	for an entity, change its class to some other random entity	[O, O, O, S-event_name, O, B-person, I-person]
	bias	for an entity, change its class to some particular entities according to a transition matrix	[O, O, O, S-event_name, O, B-time I-time]

Table 1: Summarized synthetic noise generation scheme.

the samples with a large loss, then it uses the uniform partial CRF to relearn the noisy tokens. As for alternative automatic cutoff fitting methods, we also consider replacing our FitMix with the method from (Pleiss et al., 2020) (and call it **CutFake**), which manually assigns several tokens with labels of an additional "fake" class and uses the lower tail of their confidence scores for sample selection.

Implementation details In this paper, we consider using two types of encoders: one is the BiLSTM encoder (Huang et al., 2015) and the other one is the BERT encoder (Devlin et al., 2019). For BiLSTM, we use hidden dimension 200, SGD optimizer with learning rate 0.01; for BERT, we use the default hidden dimension 768, and the default optimizer with learning rate 2e-5. We use batch size 10 for both encoders and it works well. For BiLSTM encoder, we train for 30 epochs, and for BERT, we train for 20 epochs. We split the whole dataset into train/dev/test subsets if such splitting was not provided by the original dataset, and we keep the sample size ratio of train/dev/test as 2:1:1. We output the model with the best dev F1 score.

4.3 Main results

Noise Type v.s. Noise Rate Table 2 shows how the sample selection based methods work in different synthetic problem settings. Specifically, we show the results of different methods confronting one specific type of noise respectively, to investigate our initial questions about whether methods' performances depend on noise type and noise rate. For a more realistic mixed noise, we refer to Table 3. To get a sense of upper bound performance, we also consider an oracle method called **onlyClean**, where we replace the sample selection step in the original **NLNCE** method by directly telling it which is truly clean and noisy. Here Table 2 summarizes F1 score of the **baseline**, and the differences from it of denoising methods **NLNCE**, **UseClean** and **onlyClean**. The significant positive differences are marked as green, while the significant negative ones are marked as red, and the rest are marked as grey. We can see that, after doing sample selection correctly, the current sample selection based method can indeed improve a lot over the baseline, even though it still has some gaps from the fully clean supervised case in the high noise rate regime. Overall, we can observe that sample selection based methods perform differently under different noise types and noise rates. Basically, for `over` type of noise, the baseline's performance is not influenced much. We suspect that this is due to the fact that the `over` type of noise is kind of unnatural, as it randomly selects a nonentity word and assigns a random entity to it. Such nonentity words would often be meaningless words like 'the', 'a' etc, and the CRF model can autocorrect such unnatural mistakes as it optimizes over a tag sequence as a whole. Another similar case is the `swap` noise type: where we find out that baseline can already perform relatively well compared to other noise types of similar noise rates. For the rest more natural noise types, we can observe that the effectiveness of the sample selection based idea depends more on the word-level noise rate, rather than the noise type. Specifically, it is less effective when the noise rate is low. From this reason, we can see that for `shift` and `shrink` type of noises, sample selection based methods generally do not help as much as they do in the other noise types, since `shift` and `shrink` tend to have lower word-level noise rate comparing to other noise types. In the rest of the paper, we will focus on the `miss`,

utterance level noise rate	0%	30%					100%				
Noise Type / Methods		Baseline	NLNCE	UseClean	onlyClean	*Word leve noise rate*	Baseline	NLNCE	UseClean	onlyClean	*Word leve noise rate*
Miss		76.62	+0.15	+1.03	+4.39	11%	35.26	+12.17	**+21.47**	**+37.55**	50%
Over		80.97	+0.60	-0.38	+0.43	17%	77.79	+0.45	+0.82	+3.41	41%
Shift		79.32	-0.01	+0.85	+0.93	6%	70.83	-0.40	+1.57	+8.78	20%
Extend	80.85	74.53	+0.13	+4.34	+6.84	18%	42.4	+5.56	**+24.36**	**+35.35**	52%
Shrink		77.38	-0.25	+0.66	+3.13	8%	60.49	-1.56	+2.29	+15.41	33%
Swap		78.74	+0.32	-0.51	+2.85	21%	56.33	+0.03	+0.95	+14.93	67%
Bias		75.15	-0.08	+2.05	+5.21	15%	38.75	+0.43	**+20.1**	**+33.09**	49%

Table 2: The performance of baseline, NLNCE, UseClean, onlyClean over different noise types and noise rates. The significant positive differences are marked as green, meaning that the method improves over 1% over baseline, while the significant negative ones are marked as red, meaning the method is even worse than baseline by over 1%; the rest are marked as grey. The extremely positive ones are marked in bold green. We use the BERT encoder throughout all those experiments.

`extend`, `bias` type of noises under the high noise rate regime (70%, 100% utterance level noise rate), where we know sample selection kind of idea has the potential to help much.

Broad synthetic and realistic noisy settings Table 3 summarizes the results for a more complete collection of denoising methods and more realistic noisy datasets. For all the noisy datasets, we report their summed word-level noise rate over nonentities and entities. We can see that, the word-level noise rate on the realistic noisy dataset tends to be pretty high, therefore suitable for applying our sample selection based method. We can see that, NLNCE and CoReg can improve over the baseline a bit under very limited cases, while our method UseClean can improve over the baseline by a large margin over all those noisy datasets.

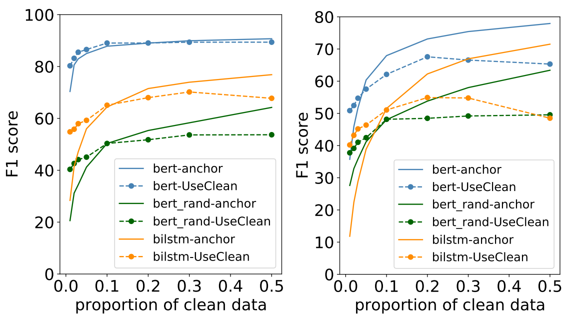
4.4 Ablation Study

	miss		extend		bias	
utterance level noise rate	70%	100%	70%	100%	70%	100%
word level noise rate	34%	50%	39%	52%	34%	49%
(oracle)	65.17	52.51	70.75	63.29	58.73	56.43
adapt (oracle)	65.17	54.44	72.15	66.40	63.03	56.43
warm (oracle)	67.58	54.21	71.63	64.92	61.14	58.35
weight (oracle)	67.31	55.08	69.52	65.15	59.12	58.28
UseClean (oracle)	**67.56**	**56.73**	**73.02**	**66.77**	**65.05**	**58.85**
UseClean (fitmix)	67.37	54.10	70.29	61.04	64.73	55.80

Table 4: Ablation Study for our method.

In Table 4 we show results for the ablation study, to see how each component in our method contributes to the final performance. Here the first line represents random sampling which is our base, and adapt means only uses adaptive sampling; warm means only uses warm start; weight means only uses weighted semi-supervised learning. For fair comparison without the confounding effect from cutoff fitting, we simply use the oracle cutoff: we fit a logistic regression of true clean/noisy labels with the confidence score, and use the predicted clean/noisy labels as sample selection decisions. We can see that each of these three techniques improves over the base in most cases, and a combination of them, which is adapt + warm + weight improves over the base by about 2-4% over all cases. Finally, our FitMix technique can achieve performance close to the oracle cutoff.

4.5 Further analysis



(a) CoNLL03 data (b) Massive data

Figure 3: The performance with and without the denoising step in UseClean versus different size of clean supervision.

Amount of clean supervision required We would like to explore how much clean data should we require such that it is reasonable to ask. To be more specific, we would like the effectiveness of our method also comes from the denoising part, rather than just the clean data pertaining part. In Figure 3 we compare the F1 score for the clean anchor model and our UseClean model with different proportion of clean dataset. To take account

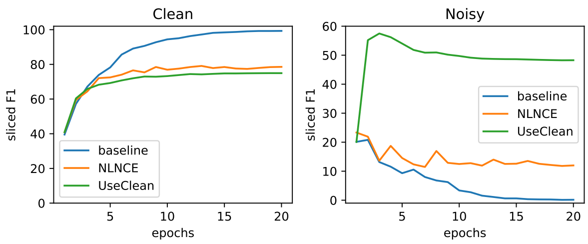
	simulated noise						realistic noise				
	miss		extend		bias		distant supervision			transfer learning	
utterance level noise rate	70%	100%	70%	100%	70%	100%	Massive	CoNLL03	Wikigold	Massive	CoNLL2003
word level noise rate	34%	50%	39%	52%	34%	49%	66%	22%	48%	50%	61%
baseline	58.8	35.26	56.38	42.4	55.04	38.75	42.02	72.76	49.76	50.7	35.88
NLNCE	64.34	47.42	62.94	47.96	58.91	39.18	40.93	72.44	54.27	51.24	42.73
NLNCE*	65.28	49.13	65.56	46.40	58.91	39.81	42.87	74.58	57.62	51.99	44.78
CoReg	48.80	38.91	47.76	41.95	45.30	37.88	41.52	70.64	49.33	52.18	34.34
CutFake	53.85	**54.47**	61.18	58.01	56.14	55.79	53.51	**79.27**	55.48	60.60	51.77
UseClean	**67.37**	54.18	**70.29**	**61.04**	**64.73**	**55.80**	**57.78**	77.31	**68.08**	**61.25**	**76.11**

Table 3: The performance of our method and all the competitors over simulated noise and realistic noise.

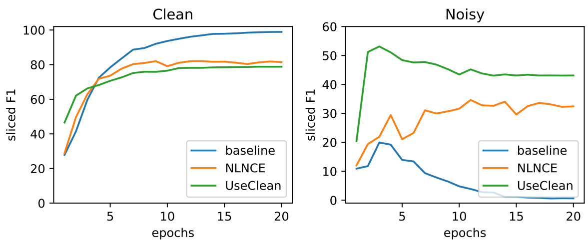
of the influence from dataset, model architecture and pretraining, we consider one simple dataset CoNLL03 and one complex data set Massive; and we consider three different backbone models: `bert` (pretrained BERT model); `bert_rand` (randomly initialized BERT model); `bilstm` (randomly initialized BiLSTM model). Due to the limitation of space, here we only demonstrate the results of one noisy type under the high noise rate regime: the bias type of noise with utterance noise level 100%.

By just looking at the solid lines (i.e. clean anchor model performance), we can see that all lines tend to first rise rapidly and then slows-down, this phenomenon is more evident on this simpler data set CoNLL03 and large pertained encoder. This indicates that large pretrained model is less data-hungry, especially in the easy problem setting. Also, we can see that, augmentation only outperforms non-augmentation when clean data is limited. Therefore, we argue that the reasonable size of clean supervision we require should be less than hundreds of examples. In all our examples, we use 100-200 examples depending on the problem difficulty.

Training dynamics In Figure 4, we plot the sliced F1 score on clean tokens and noisy tokens and also the total F1 score during the whole training process for baseline, NLNCE and UseClean. We can see that, both NLNCE and UseClean can indeed learn on the noisy tokens, while UseClean tends to learn much better on the noisy case without sacrificing too much on the clean cases. For the case where UseClean is much better than NLNCE (i.e. Figure 4(a)), its generalization gap is the smallest among the three methods. For the case where UseClean is a bit better than NLNCE (i.e. Figure 4(b)), we find out that the generalization gap of both NLNCE and UseClean is nearly none, meaning that now NLNCE also does not overfit to noise too much. Still, UseClean learns better on noisy cases. Finally, we also find out that, the sample selection type of denoising method improves over baseline mainly by learning better on the noisy cases, which often at the cost of a certain amount of performance drop on clean cases. This might also explain our findings about why the sample selection type of idea only works in a high noise rate regime.



(a) Bias type noise



(b) Miss type noise

Figure 4: The sliced F1 score on clean tokens and noisy tokens during the whole training process.

5 Conclusion and Discussion

In conclusion, we propose an effective and efficient method called UseClean, which includes a simple two-stage framework to avoid error accumulation, a novel confidence score utilizing minimal clean supervision to increase predictive power in sample selection, an automatic cutoff fitting to save extensive hyper-parameter tuning and finally weighted semi-supervised learning with warm start to learn better on the noisy tokens. Additionally, we propose to use adaptive sampling to construct better clean supervision for a further performance boost. Despite simple, our method improves F1 score by on average at least $5 \sim 10\%$ over current state-of-art without extensive hyper-parameter tuning or heavy computation, and is effective across a broad type of noise types and noise levels.

We admit that most of the performance gain comes from the minimal clean supervision in small gold data. Without it, the SOTA method NLNCE suffers from error accumulation and heavy computation. Still, we argue that the clean supervision we need is very minimal, like just about 100 samples, while the stable improvement and efficiency it can bring is fairly large. In fact, we suspect that it is often necessary to guarantee success in real applications, and how to best construct and leverage clean supervision is nontrivial and important.

Limitations

We admit that the methods for comparison in this paper are not a complete list of the literature, though we arguably claim that they are strong representatives. We omit some methods for now due to their complexity and computation time. It would make our paper a more convincing story if we had also considered the rest established methods like **BOND**(Liang et al., 2020). Also, currently we do the sample selection and semi-supervised learning in a one-pass way, while alternatively an iterative-pass way like active learning (Kong et al., 2021) might be even more effective. Still, one need to be careful about the error propagation during the iterative process.

Even though we point out the importance and potential of designing and leveraging the meta gold dataset, we have not provided a thorough discussion of past endeavors. Particularly, **FilDist** (Onoe and Durrett, 2019) also utilize clean supervision like us, though they also require corresponding noisy labels to fit a binary classifier for sample selection. It would be interesting to see how those methods compare to ours.

References

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR.

Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R. Curran. 2009. Named entity recognition in Wikipedia. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources (People's Web)*, pages 10–18, Suntec, Singapore. Association for Computational Linguistics.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32.

Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. 2017. Active bias: Training more accurate neural networks by emphasizing high variance samples. *Advances in Neural Information Processing Systems*, 30.

Xinlei Chen and Abhinav Gupta. 2015. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1431–1439.

Ziheng Chen, Fabrizio Silvestri, Jia Wang, Yongfeng Zhang, and Gabriele Tolomei. 2023. The dark side of explanations: Poisoning recommender systems with counterfactual examples. *arXiv preprint arXiv:2305.00574*.

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.

Guimin Dong, Mingyue Tang, Lihua Cai, Laura E Barnes, and Mehdi Boukhechba. 2021. Semi-supervised graph instance transformer for mental health inference. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1221–1228. IEEE.

Anqi Feng, Yuan Xue, Yuli Wang, Chang Yan, Zhangxing Bian, Muhan Shao, Jiachen Zhuo, Rao P Gullapalli, Aaron Carass, and Jerry L Prince. 2023. Label propagation via random walk for training robust thalamus nuclei parcellation model from noisy annotations. *arXiv preprint arXiv:2303.17706*.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, et al. 2022. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. *arXiv preprint arXiv:2204.08582*.

Jacob Goldberger and Ehud Ben-Reuven. 2017. Training deep neural-networks using a noise adaptation layer. In *International Conference on Learning Representations*.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31.

Yunzhong He, Cong Zhang, Ruoyan Kong, Chaitanya Kulkarni, Qing Liu, Ashish Gandhe, Amit Nithianandan, and Arul Prakash. 2023. Hiercat: Hierarchical query categorization from weakly supervised data at facebook marketplace. In *Companion Proceedings of the ACM Web Conference 2023*, pages 331–335.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. pages 2304–2313. PMLR.

Zhanming Jie, Pengjun Xie, Wei Lu, Ruixue Ding, and Linlin Li. 2019. Better modeling of incomplete annotations for named entity recognition. 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 729–734.

Ruoyan Kong, Zhanlong Qiu, Yang Liu, and Qi Zhao. 2021. Nimblelearn: A scalable and fast batch-mode active learning approach. In *2021 International Conference on Data Mining Workshops (ICDMW)*, pages 350–359. IEEE.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Ji Young Lee, Franck Dernoncourt, and Peter Szolovits. 2017. Transfer learning for named-entity recognition with neural networks. *arXiv preprint arXiv:1705.06273*.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020a. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Junnan Li, Richard Socher, and Steven CH Hoi. 2020b. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*.

Xiaomin Li and Vangelis Metsis. 2022. Spp-eegnet: An input-agnostic self-supervised eeg representation model for inter-dataset transfer learning. In *International Conference on Computing and Information Technology*, pages 173–182. Springer.

Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. Bond: Bert-assisted open-domain named entity recognition with distant supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1054–1064.

Kun Liu, Yao Fu, Chuanqi Tan, Mosha Chen, Ningyu Zhang, Songfang Huang, and Sheng Gao. 2021. Noisy-labeled ner with confidence estimation. *arXiv preprint arXiv:2104.04318*.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. 2021. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*.

Aditya Krishna Menon, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. 2020. Can gradient clipping mitigate label noise? In *International Conference on Learning Representations*.

Curtis Northcutt, Lu Jiang, and Isaac Chuang. 2021. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411.

Yasumasa Onoe and Greg Durrett. 2019. Learning to denoise distantly-labeled data for entity typing. 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 2407–2417.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. 2020. Identifying mislabeled data using the area under the margin ranking. In *Advances in Neural Information Processing Systems*, volume 33, pages 17044–17056. Curran Associates, Inc.

Jayaram Raghuram, Yijing Zeng, Dolores Garcia, Rafael Ruiz, Somesh Jha, Joerg Widmer, and Suman Banerjee. 2022. Few-shot domain adaptation for end-to-end communication. In *The Eleventh International Conference on Learning Representations*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE*

Transactions on Neural Networks and Learning Systems.

Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. 2015. Training convolutional networks with noisy labels. In *3rd International Conference on Learning Representations, ICLR 2015*.

Charles Sutton, Andrew McCallum, et al. 2012. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4):267–373.

Yuta Tsuboi, Hisashi Kashima, Shinsuke Mori, Hiroki Oda, and Yuji Matsumoto. 2008. Training conditional random fields using incomplete annotations. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 897–904.

Ruxin Wang, Tongliang Liu, and Dacheng Tao. 2017. Multiclass learning with partially corrupted labels. *IEEE transactions on neural networks and learning systems*, 29(6):2568–2580.

Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. 2018. Iterative learning with open-set noisy labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8688–8696.

Yuli Wang, Ryan Herbst, and Shiva Abbaszadeh. 2021. Electronic noise characterization of a dedicated head-and-neck cancer pet based on czt.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. Crossweigh: Training named entity tagger from imperfect annotations. *arXiv preprint arXiv:1909.01441*.

Hongxin Wei, Lue Tao, RENCHUNZI XIE, and Bo An. 2021. Open-set label noise can improve robustness against inherent label noise. In *Advances in Neural Information Processing Systems*.

Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. 2021. Robust early-learning: Hindering the memorization of noisy labels. In *International Conference on Learning Representations*.

Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. 2019. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, pages 7164–7173. PMLR.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021a. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.

Dan Zhang and Fangfang Zhou. 2023. Self-supervised image denoising for real-world images with context-aware transformer. *IEEE Access*, 11:14340–14349.

HaiYang Zhang, XiMing Xing, and Liang Liu. 2021b. Dualgraph: A graph-based method for reasoning about label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9654–9663.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Wenqing Zheng, Edward W Huang, Nikhil Rao, Sumeet Katariya, Zhangyang Wang, and Karthik Subbian. 2021. Cold brew: Distilling graph node representations with incomplete or missing neighborhoods. *arXiv preprint arXiv:2111.04840*.

Tianyi Zhou, Shengjie Wang, and Jeff Bilmes. 2020. Robust curriculum learning: from clean label detection to noisy label self-correction. In *International Conference on Learning Representations*.

Wenxuan Zhou and Muhao Chen. 2021. Learning from noisy labels for entity-centric information extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5381–5392, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Benchmarking Neural Network Generalization for Grammar Induction

Nur Lan[1,2], Emmanuel Chemla[1], Roni Katzir[2]
[1]Ecole Normale Supérieure
[2]Tel Aviv University
{nur.lan,emmanuel.chemla}@ens.psl.eu
rkatzir@tauex.tau.ac.il

Abstract

How well do neural networks generalize? Even for grammar induction tasks, where the target generalization is fully known, previous works have left the question open, testing very limited ranges beyond the training set and using different success criteria. We provide a measure of neural network generalization based on fully specified formal languages. Given a model and a formal grammar, the method assigns a generalization score representing how well a model generalizes to unseen samples in inverse relation to the amount of data it was trained on. The benchmark includes languages such as $a^n b^n$, $a^n b^n c^n$, $a^n b^m c^{n+m}$, and Dyck-1 and 2. We evaluate selected architectures using the benchmark and find that networks trained with a Minimum Description Length objective (MDL) generalize better and using less data than networks trained using standard loss functions. The benchmark is available at https://github.com/taucompling/bliss.

1 Introduction

The extent to which artificial neural networks (ANNs) generalize beyond their training data is an open research question. In this work we approach this question from the perspective of grammar induction, that is, the learning of a formal grammar from a finite (often small) sample from the (typically infinite) language of that grammar. In order to succeed in this task, a model must strike a balance between fitting the training data and generalizing to a potentially infinite set of unseen strings. Humans tested on such tasks show systematic generalization from small sets of examples (Fitch and Hauser, 2004, Malassis et al., 2020).

While a range of ANN architectures have been shown to reach approximations for formal languages, the quality of this approximation remains an open matter, as we show below. Here we build on previous probes of ANN generalization for grammar induction and introduce a unified and general way to assess this capability, for a given pair of a learning model and a corpus drawn from a formal language. Our main contributions are:

1. **A benchmark for formal language learning**. The benchmark relies on a method for quantifying ANN generalization for formal languages, including probabilistic languages. The method assigns an index score representing a model's generalization performance in inverse relation to the size of the training data. We introduce the method and provide concrete datasets for well-studied formal languages.

2. **An evaluation of selected architectures.** We test recurrent neural networks (RNNs) of the Long-Short Term Memory type (LSTM; Hochreiter and Schmidhuber, 1997); Memory-augmented RNNs (MARNN; Suzgun et al., 2019b;) and an RNN variant which replaces the traditional gradient-based training regime with an objective that optimizes the model's Minimum Description Length (MDLRNN; Lan et al., 2022).

   We find that equipping ANNs with memory devices such as differentiable stacks helps generalization, but generalization remains partial and slow. At the same time, training with MDL leads in some of the test cases that we examined to potentially perfect generalization with significantly less data. In other cases, training with MDL did not result in successful generalization, possibly because the optimization procedure we used for the architecture search failed to find the global optimum under the MDL objective function.

2 Background

Learning formal languages has long been used to probe various aspects of ANN performance. These most often include inquiries about: (i) ANNs' ability to generalize beyond their training data, and

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 131–140
September 11–12, 2023.

Language	Paper	Model	Metric	Training size	Max train n	Max test n
a^nb^n	GS'01	LSTM	$M_{cat'}$	16,000	30	1,000
	JM'15	Stack-RNN	M_{det}	$20^{\dagger}$	19	60
	WGY'18	LSTM	Bin	$100^{\dagger}$	100	256
	LGCK'22	MDLRNN	M_{det}	500	22	∞
$a^nb^nc^n$	GS'01	LSTM	$M_{cat'}$	51,000	40	500
	JM'15	Stack-RNN	M_{det}	$20^{\dagger}$	19	60
	WGY'18	LSTM	Bin	$50^{\dagger}$	50	100
	LGCK'22	MDLRNN	M_{det}	500	22	∞
Dyck-1	SGBS'19a	LSTM	$M_{cat'}$	10,000	50	100
	SGBS'19b	MARNN	$M_{cat'}$	5,000	50	100
	EMW'22	ReLU-RNN	$M_{cat'}$	10,000	50	1,000
	LGCK'22	MDLRNN	M_{cat}	500	16	∞

Table 1: ANN performance in selected probes of formal language learning. **Metrics** (see Section 3.5): M_{det} = deterministic accuracy; M_{cat} = categorical accuracy; $M_{cat'}$ = a non-probabilistic version of M_{cat}; Bin = binary classification from hidden state to accept/reject labels, based on positive and negative samples. **Training size**: † = the paper did not explicitly specify the training set size, the value here is derived by assuming the training set was an exhaustive list of all strings up to 'max train n'. **'Max test n'**: the largest n for which the criterion was reached. For Dyck-1, n represents overall sequence length. '∞' = the paper provides evidence that the network is correct for any n. When a paper reports several experiments as in GS'01, we take the best result based on the smallest training set. **Papers**: GS'01 = Gers and Schmidhuber (2001); JM'15 = Joulin and Mikolov (2015); WGY'18 = Weiss et al. (2018); SGBS'19a = Suzgun et al. (2019a); SGBS'19b = Suzgun et al. (2019b); EMW'22 = El-Naggar et al. (2022); LGCK'22 = Lan et al. (2022).

(ii) ANNs' expressive power; i.e., whether they can represent the relevant target grammars (often probed with reference to the Chomsky hierarchy of formal languages, as in Delétang et al., 2022). Here we will focus on the generalization question. We will show how it might be related to another under-exploited line of inquiry regarding the training objective of ANNs.

A long line of theoretical work has probed the computational power of ANNs. Siegelmann and Sontag (1992) originally showed that RNNs with a sigmoid activation can emulate multiple-stack Turing machines under certain permissive conditions (infinite activation precision and unbounded running time). Since these conditions cannot be met in practice, another line of work probed the computational power of RNNs under practical conditions (finite precision and input-bound running time). Weiss et al. (2018) have shown that under these conditions LSTMs are able to hold weight configurations that perform unbounded counting, and so they should be able to recognize counter languages (CL), a family of formal languages that can be recognized using one or more counting devices (following some formal restrictions, Merrill, 2021). Recently, El-Naggar et al. (2023a) and El-Naggar et al. (2023b) have shown that two simpler RNN architectures, with linear- and ReLU-based cells, are also able to hold counting weight configurations, with similar consequences for recognizing CL.

Empirically, another line of work provided promising results regarding the capability of ANNs to learn formal languages. This was most often done by training networks on strings up to a certain length and then showing good performance on longer ones (Bodén and Wiles, 2000, Gers and Schmidhuber, 2001; see Table 1). Gers and Schmidhuber (2001) have shown that LSTMs trained on languages such as a^nb^n and $a^nb^nc^n$ with n values in the low dozens perform well on n's in the high hundreds. Suzgun et al. (2019a) found that LSTMs trained on Dyck-1 sequences (strings of well-balanced pairs of brackets) up to length 50 performed well on lengths up to 100. Suzgun et al. (2019b) proposed RNN variants that are equipped with external differentiable memory devices and showed that they yield improved performance on non-regular languages.

However, other empirical results show that in practice ANNs generalize only to very restricted ranges. Weiss et al. (2018) found that while LSTMs are theoretically able to hold counting solutions, these are not found through training: LSTMs trained on a^nb^n and $a^nb^nc^n$ with max n 100 and

50, respectively, start accepting illicit strings with n values as low as 256 and 100. As mentioned above, Suzgun et al. (2019a) tested LSTMs on Dyck-1 sequences but only up to length 100, and concluded that this language was learned by LSTMs. El-Naggar et al. (2022) extended this work to longer sequences, and found that LSTMs fail to generalize in practice, outputting incorrect predictions at lengths under 1,000. This, despite Dyck-1 being a CL and so theoretically learnable by LSTMs (Weiss et al., 2018).

Apart from LSTMs, recent probes by El-Naggar et al. (2023a) and El-Naggar et al. (2023b) have shown that linear and ReLU RNNs, theoretically capable of counting, fail to find the counting weight configurations in practice when trained using back-propagation and standard loss functions; El-Naggar et al. (2023b) went further with determining the source of this discrepancy, showing that the counting weight configuration is not an optimum of these loss functions.

Moreover, even in works that report successful generalization to some degree beyond the training set, the fact that networks stop generalizing at an arbitrary point is often left unexplained (Gers and Schmidhuber, 2001, Suzgun et al., 2019a, 2019b, Delétang et al., 2022, a.o.).[1]

The literature on the generalization abilities of ANNs has made use of a range of measures of success, making results difficult to compare. Different probes of the same model often use different success criteria, and generate training and test sets using different sampling methods and of different orders of magnitude. Table 1 summarizes selected probes of ANN generalization and highlights the fragmented nature of this literature. In the following sections we propose a unified method to consolidate these efforts and better understand the generalization capabilities of ANNs.

3 The BLISS index

We present the Benchmark for Language Induction from Small Sets (BLISS). We provide a formal description of the method, followed by a concrete application to specific tasks.

[1]Technical limitations such as finite activation precision can be ruled out as explanations for generalization failures, at least for counter languages and models where network states serve as memory: as shown in works mentioned above, ANNs often start outputting wrong predictions for n values in the low hundreds. Even restricted representations such as 16-bit floats can hold much larger values, and modern implementations such as PyTorch use 32-bit floats by default.

The current release consists of three parts: (i) A specification for the generalization index $\mathbb{B}$, calculated for a given pair of formal language and ANN; (ii) A dataset containing a set of formal languages for benchmarking; (iii) An evaluation of different ANN architectures using this dataset.

3.1 General setting: models and tasks

For a given model A, e.g., an LSTM, a task is composed of the following components:

- G – a grammar, e.g., a probabilistic context-free grammar (PCFG).
- S – a sampling method from $\mathcal{L}(G)$, the language generated by G.
- $\mathcal{C} = S(G)$ – a training corpus, may contain repetitions.
- $\mathcal{T} \subseteq \mathcal{L}(G) \setminus \mathcal{C}$ – a test corpus.
- M – a task-specific accuracy metric with adjustable error margin

  $\varepsilon \in [0, 1]$. It uses predictions $A(s)$ on strings $s \in \mathcal{T}$ to calculate an accuracy score $M(A, \mathcal{T}, \varepsilon) \in [0, 1]$.
- N – a task-specific constant for setting the order of magnitude of dataset sizes. For example, $N = 3$ sets the order of magnitude at 10^3. Training and test sizes are then derived as described below. Selecting N is done empirically based on properties of the task, e.g., languages with large vocabularies require larger amounts of training data, hence a larger N.

3.2 From task to generalization index

For a given task, the generalization index of order N for a model A is then defined as:

$$\mathbb{B}_N^{\mathcal{L}}(A) = max\left\{ b \;\middle|\; \begin{array}{c} |\mathcal{T}| = 10^N \times b, \\ |\mathcal{C}| = 10^N / \, b, \\ M(A, \mathcal{T}, \varepsilon) = 1.0 \end{array} \right\} \quad (1)$$

Intuitively, the index compares a model's performance on a test size $|\mathcal{T}|$ to the inverse of its training data size $|\mathcal{C}|$.

The index is expressed as the maximal b factor which scales the training set and the corresponding test set in opposite directions: The accuracy condition at the bottom of (1) means that the model should be ε-close to perfect generalization on the test set. A model's generalization index $\mathbb{B}$ thus represents the performance that can be maximally 'squeezed out' of an inversely small amount of data.

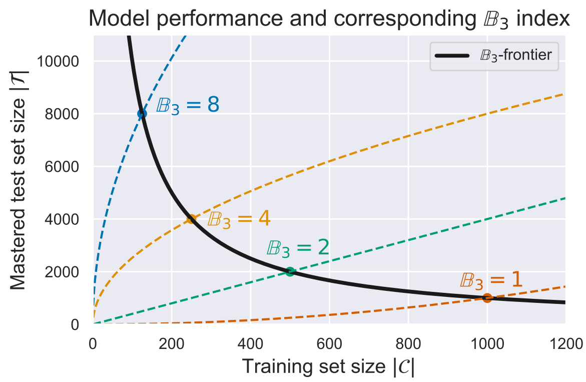


Figure 1: Example generalization index scores $\mathbb{B}_3$, i.e., for a baseline training size of 10^3. Each dashed line represents the performance profile of some hypothetical model, as a function of the size of the training set. The intersection with the $\mathbb{B}_3$-frontier indicates its $\mathbb{B}_3$ index.

Figure 1 exemplifies selected $\mathbb{B}$ values calculated based on (1). For illustration, for $a^n b^n$, using the order of magnitude $N = 3$, a model that was trained on $|\mathcal{C}| = 10^3/2 = 500$ samples and was 100% accurate on a test set of size $10^3 \times 2 = 2,000$ will have an index score $\mathbb{B}_3^{a^n b^n} \geq 2$. A model for the same language that was trained on 250 samples only and generalized to a subsequent set of 4,000 samples will reach $\mathbb{B}_3^{a^n b^n} \geq 4$.

For practical reasons, one cannot exhaust all values of b to find $\mathbb{B}$. However, training and evaluating a model using a few b values is enough to reveal its generalization dynamics, as shown in experiments in Section 5. The following sections describe the specific choices made for the different benchmark components in these experiments.

3.3 Learning setup

Previous work surveyed here differed in their learning setup. Gers and Schmidhuber (2001) and Suzgun et al. (2019a, 2019b) trained networks in a non-probabilistic, supervised setup by exposing the model to all possible next symbols and minimizing the mean-squared error (MSE) – i.e., the model is given explicit information about the distribution of possible symbols. Joulin and Mikolov (2015) and Lan et al. (2022) used a setup that we adopt below, in which model outputs are probabilistic, and training is self-supervised language modeling (i.e., the model is exposed to the next symbol only) with a cross-entropy loss. Weiss et al. (2018) trained a binary classifier with accept/reject labels based on positive and negative examples.

Since our focus is grammar induction, here we adopt the more demanding setup of learning from positive examples alone. All tasks are thus designed as self-supervised language modeling. At each time step, a model assigns a probability distribution to the next symbols in the string.

The benchmark is agnostic as to the internals of the model and its training, as long as its outputs represent a probability distribution over symbols. In practice, then, the method can be applied to any language model, not necessarily an ANN.

3.4 Sampling

To construct the training and test sets $\mathcal{C}$ and $\mathcal{T}$ we use the following as method S:

- To construct $\mathcal{C}$, we sample strings according to the distribution defined by G, with repetitions. For example, if G is a PCFG, it can be sampled by applying derivation rules chosen proportionally to their expansion probabilities. Repetitions are allowed so that $\mathcal{C}$ follows a similar surface distribution to $\mathcal{L}(G)$ and so that the model can pick up on the underlying probabilities in G.

- To construct $\mathcal{T}$, we take the $|\mathcal{T}|$ subsequent strings starting right after the longest string in $\mathcal{T}$, sorted by length.[2] For example, for the language $a^n b^n$, if the longest string in the training set $\mathcal{C}$ was $a^{17}b^{17}$, and the model needs to be tested on a set of 2000 strings, $\mathcal{T}$ will be composed of the strings $a^{18}b^{18}, \ldots, a^{2017}b^{2017}$.

The sampling method S can be either probabilistic as described here, or exhaustive, training on all strings in $\mathcal{L}$ up to a certain length. We opt for probabilistic sampling because of the nature of the task at hand: the models under discussion here are trained to assign probabilities to the next symbol in a string, most often minimizing a cross-entropy loss. In practice, then, they always learn distributions over strings. Thus if $\mathcal{C}$ follows a similar surface distribution to $\mathcal{L}$ (given a large enough sample size), the model should eventually learn this distribution in order to minimize its loss.

Probabilistic sampling thus makes it possible to probe both a model's knowledge about the surface forms of $\mathcal{L}$ (by treating model outputs as categorical classes), and about their distribution. The modularity of the index makes it possible to choose

[2]Test strings may need to be sorted further according to specific properties of a language, see Section 4.1.

either option by varying the accuracy metric M, as we show in the next section.

3.5 Accuracy metrics

Ultimately we are interested in knowing whether a model accepts all strings in $\mathcal{L}$ and rejects all others. In classical formal language theory, where discrete automata are used, acceptance is clear cut and taken as going into an accepting state. ANNs on the other hand use continuous representations with no standard acceptance criterion.

Different acceptance criteria have been used in previous works to measure success for ANNs: Gers and Schmidhuber (2001) and Suzgun et al. (2019b) defined acceptance of a string as a model assigning output values above a certain threshold to valid symbols only; Joulin and Mikolov (2015) measure accuracy at parts of strings that are completely predictable; and Weiss et al. (2018) turn a network into a recognizer by training a binary classifier from network states to accept/reject labels. Below we provide general versions of these accuracy metrics (omitting Weiss et al., 2018 who rely on negative examples).

Choosing which metric to use is based on the properties of the language at hand. Well-performing models might still deviate slightly from perfect accuracy due to practical limitations, such as a softmax function preventing a model from expressing categorical decisions. Thus for each accuracy metric we add an adjustable error margin ε. Acceptance of a string is defined as reaching 100% accuracy (minus ε) on the string. Success on the test set is then defined as accepting all strings in the set (third condition in (1)).

Input:	#	(	(	)	(	)	)
	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
Target:	#/(	(/)	(/)	(/)	(/)	(/)	#/(

Input:	#	a	a	a	b	b	b
	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
Target:	$\#/a$	a/b	a/b	a/b	$\boldsymbol{b}$	$\boldsymbol{b}$	**#**
						Deterministic	

Figure 2: Inputs and valid next symbols at each step of a Dyck-1 string (top) and a^nb^n (bottom), including the start/end-of-sequence symbol '#'. For a^nb^n, accuracy is measured at deterministic steps, after the first 'b'. For Dyck-1, accuracy is the fraction of time steps where a model predicts only valid next symbols: '#' should be predicted only when brackets are well balanced.

1. *Deterministic accuracy* (M_{det}). Some languages contain strings with deterministic phases, where the next symbol is fully predictable. For example, strings in the language a^nb^n have two phases, the a phase and the b phase. As long as only a's are seen, the next symbol remains unpredictable as the sequence can continue with another a or switch to the b phase. The string becomes deterministic once the first b appears. M_{det} is defined as the fraction of deterministic time steps in which the model assigns the majority probability to the correct next symbol. This metric is used in Joulin and Mikolov (2015).

   A string is considered accepted if the model is $1-\varepsilon$ accurate over all deterministic time steps. Note however that even a very small ε might benefit models that do not recognize strings well. For example, for the language a^nb^n, the deterministic steps in a string are the b's and the final end-of-sequence symbol. A degenerate model that predicts only b's will get only the end-of-sequence symbol wrong out of all deterministic steps, and will reach a very high accuracy score. For any large enough test set these errors will be hidden within the ε margin and the model will be deemed successful. ε should therefore be chosen with care per task.

   M_{det} is used below for the following languages that have deterministic phases: a^nb^n, $a^nb^nc^n$, $a^nb^nc^nd^n$, and $a^nb^mc^{n+m}$.

2. *Categorical accuracy* (M_{cat}). Some language strings do not have any predictable phases. This is the case in the Dyck family of languages. At each time step in a Dyck string, one may open a new bracket (see Figure 2). M_{cat} is therefore defined as the fraction of steps in which a network assigns probability $p > \varepsilon$ to each possible next symbol, and $p \leq \varepsilon$ to irrelevant symbols. Non-probabilistic versions of M_{cat} are used in Gers and Schmidhuber (2001) and Suzgun et al. (2019a, 2019b) who do not treat network outputs as probability distributions. M_{cat} is used below for Dyck languages.

As specified in Section 3.1, the index $\mathbb{B}$ is calculated based on the largest test set for which a model reaches an ε-perfect accuracy score.

Beyond accuracy, one might be interested in inspecting a model's knowledge of the distribution of strings in $\mathcal{L}$ induced by a probabilistic G. This can be done by using the probabilistic sampling method described in Section 3.4 and accompanying it with a probabilistic accuracy measure – for example, one based on an optimal cross-entropy score, which is known from G's expansion probabilities (as done in Lan et al., 2022). Feeding loss values into an accuracy metric will require normalizing them across tasks. We leave this extension for future work.

3.6 String structure

Following Gers and Schmidhuber (2001), each sequence starts and ends with a start/end-of-sequence symbol '#'. This turns the task into a strict acceptance/rejection task – predicting the end-of-sequence symbol is taken as going into an accept state. The start- and end-of-sequence symbols are added to the task-specific vocabulary and are assigned probabilities by the model at each step. Figure 2 illustrates input and target sequences for a^nb^n and Dyck-1.

3.7 Limitations

One shortcoming of the proposed index score is that it does not reflect perfect generalization, i.e., it is an empirical index that cannot point out a model that outputs correct predictions for *any* string in $\mathcal{L}(G)$. For most models, this will not be a problem, and $\mathbb{B}$ will simply represent the model's best training vs. test size ratio. In the case of a model that reaches perfect generalization on any input, the index score will represent the critical training size that brings the model to this performance.

Assigning a generalization score to infinitely correct models will remain a problem for any empirical metric that assigns scores to models based on finite test values. An alternative to such empirical probes would be to analytically show that a model is correct (as done in Lan et al., 2022).

4 Datasets

We provide training and test datasets for a preliminary set of formal languages for evaluation using the $\mathbb{B}$ index. The dataset includes the languages a^nb^n, $a^nb^nc^n$, $a^nb^nc^nd^n$, $a^nb^mc^{n+m}$, Dyck-1, and Dyck-2. The source code, datasets, and specifications for the benchmark are available at https://github.com/taucompling/bliss.

4.1 Training and test sets

Training sets for context-free languages are sampled from PCFGs as described in Section 3.4. The PCFGs are given in Appendix B. Training sets for context-sensitive languages are generated by sampling values for n from a geometric distribution.

Test sets are generated using the method described in Section 3.4: All test sets consist of an exhaustive list of strings ordered by length starting right after the longest string seen during training. Test sets for $a^nb^mc^{n+m}$ consist of the list of strings starting after the last seen pair of n, m, sorted by $n+m$ values to test all possible combinations.

5 Experiments

5.1 Models

We test the following models: LSTM RNNs (Hochreiter and Schmidhuber, 1997); Memory-augmented RNNs (MARNN; Suzgun et al., 2019a); and Minimum Description Length RNNs (MDL-RNN; Lan et al., 2022).

LSTM architectures were developed with the task of keeping items in memory over long distances in mind. As mentioned above, Weiss et al. (2018) have shown that LSTMs are theoretically capable of recognizing CL.

MARNNs (Suzgun et al., 2019b) are RNNs equipped with external memory devices, and were shown to yield better performance when learning languages that require stack-like devices and beyond. Here we use Stack-LSTM, an LSTM augmented with a pushdown automaton; and Baby Neural Turing Machines (Baby-NTM; itself a variant of NTMs, Graves et al., 2014), an RNN with a more freely manipulable memory.[3]

MDLRNNs are RNNs trained to optimize the Minimum Description Length objective (MDL; Rissanen, 1978), a computable approximation of Kolmogorov complexity, the algorithmic complexity of a model. The intuition behind the objective is equating compression with finding regularities in the data: a model that compresses the data well will generalize better and avoid overfitting. In practice, optimization is done by minimizing the sum of

[3]We modify Suzgun et al. (2019a)'s models to output probability distributions, replacing the final sigmoids with a softmax layer and the MSE loss with cross-entropy. See Section 3.3.

the architecture encoding length and the standard cross-entropy loss, both measured in bits based on a specific encoding scheme.

MDL is a stricter regularizer than standard regularization techniques such as L1/L2: the latter penalize large weight values but cannot prevent models from overfitting using a solution that uses many small, but informative, weights. MDL penalizes the actual information content of the network, forcing it to be general and avoid overfitting. MDLRNNs were shown to learn some of the languages discussed here in full generality using small architectures of only 1 or 2 hidden units and to outperform L1/L2 (Lan et al., 2022).

MDL is a non-differentiable objective, which requires that MDLRNN be optimized using a non-gradient based search method, such as an evolutionary algorithm that searches the network architecture space. Since this method is not based on gradient descent, Lan et al. (2022) were able to use non-standard, non-differentiable activations such as step functions. Here we restrict the architecture space to only standard activations: the linear function, ReLU, and tanh. This serves both to compare MDLRNN with standard networks and to limit the architecture search space. We publish the resulting nets as part of the MDLRNN-Torch release at https://github.com/0xnurl/mdlrnn-torch.

Appendix A lists the hyper-params for all runs.

5.2 Training sets

We used training sizes $|\mathcal{C}| = 100, 250, 500, 1000$. We stopped at the smallest size 100 because in our setup this size results in test strings of lengths $> 10{,}000$, leading to very long running times.

5.3 Index parameters

We calculate the $\mathbb{B}$ index for all trained networks using the following index parameters:

Magnitude parameter $N = 3$, i.e., training and test sizes are derived from a baseline size 10^3. This order of magnitude was selected based on the training set sizes used in previous works for the languages inspected here (Table 1).

M_{det} $\varepsilon = 0.005$, i.e., a model needs to correctly predict the next symbol for at least 99.5% of all deterministic steps. Since even this high threshold allows a degenerate model to reach good scores as described in Section 3.5, we also calculate the index score using $\varepsilon = 0$, i.e. a model must predict *all* deterministic symbols correctly.

Language	Model	$\mathbb{B}$-index $\varepsilon = 0.005$	$\mathbb{B}$-index $\varepsilon = 0$
$a^n b^n$	LSTM	**10**	<1
	Stack-LSTM	**10**	<1
	Baby-NTM	**10**	1
	MDLRNN	**10**	**10**
$a^n b^n c^n$	LSTM	<1	<1
	Stack-LSTM	2	<1
	Baby-NTM	**10**	<1
	MDLRNN	<1	<1
$a^n b^n c^n d^n$	LSTM	<1	<1
	Stack-LSTM	1	<1
	Baby-NTM	**4**	<1
	MDLRNN	<1	<1
$a^n b^m c^{n+m}$	LSTM	<1	<1
	Stack-LSTM	**10**	<1
	Baby-NTM	4	<1
	MDLRNN	4	**4**
Dyck-1	LSTM	<1	<1
	Stack-LSTM	<1	<1
	Baby-NTM	<1	<1
	MDLRNN	**2**	**2**
Dyck-2	LSTM	<1	<1
	Stack-LSTM	<1	<1
	Baby-NTM	<1	<1
	MDLRNN	<1	<1

Table 2: Generalization scores $\mathbb{B}$. The index represents how well a model generalizes in relation to its training size. A score $\mathbb{B} = 4$ indicates that a model trained on 250 samples reached the accuracy criterion on the consecutive 4,000 unseen test samples. $\mathbb{B} < 1$ indicates that the model did not reach the accuracy criterion when the test size was greater than the training size, but might reach it for larger training and smaller test sets.

M_{cat} $\varepsilon = 0.005$, i.e., for Dyck, a model needs to assign $p \leq 0.005$ to each irrelevant symbol and $p > 0.005$ to possible ones. Here as well we report results for $\varepsilon = 0$, i.e., a model must assign non-zero probabilities to valid symbols only.

6 Results

6.1 Non-perfect accuracy

The generalization index obtained by each model for each language is presented in Table 2.

We start by inspecting the indexes calculated using the more lenient accuracy margin $\varepsilon = 0.005$.

For $a^n b^n$, under this accuracy margin, all models are assigned index $\mathbb{B} = 10$, i.e., reaching the success criterion for the next unseen 10,000 samples

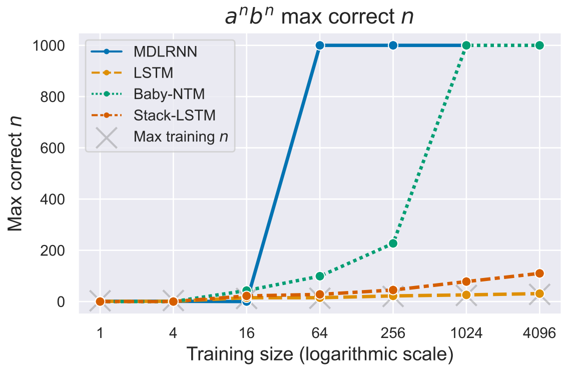


Figure 3: Generalization performance of the models tested here. Models were trained on strings drawn from $a^n b^n$ and tested on acceptance of strings up to n = 1,000. X's mark the maximum n seen during training.

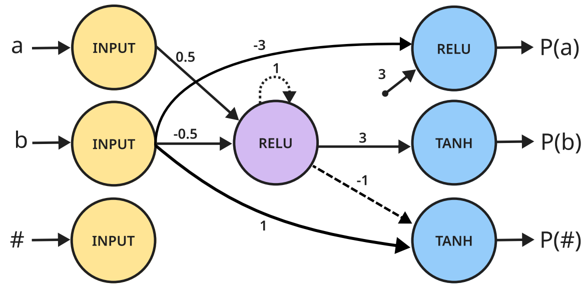


Figure 4: RNN cell architecture of the best-performing MDLRNN for $a^n b^n$, which trained on 100 samples and reached $\mathbb{B}_3 = 10$. The network uses only one hidden unit and standard activation functions, and generalizes up to at least $a^{35000}b^{35000}$. Dashed arrows are recurrent connections across time steps. The loop from the hidden ReLU unit to itself is a counter mechanism evolved by the evolutionary algorithm to count and compare the number of a's and b's.

after being trained on 100 samples. For the specific combination of random seed and sampling prior in these experiments, this means that the models were trained on strings up to $a^{20}b^{20}$ and generalized to all strings up to at least $a^{10020}b^{10020}$ with deterministic accuracy $M_{det} \geq 99.5\%$.

For $a^n b^n c^n$, MARNNs reach $\mathbb{B}$ = 10 and 2, while LSTM and MDLRNNs do not reach the success criterion, resulting in $\mathbb{B} < 1$. For $a^n b^n c^n d^n$ only MARNNs reach a specified index, with a Baby-NTM reaching $\mathbb{B} = 4$, indicating that it generalized to strings as long as $a^{4020}b^{4020}c^{4020}d^{4020}$ with $M_{det} \geq 99.5\%$.

For the addition language $a^n b^m c^{n+m}$, Stack-LSTM and MDLRNN reached index scores $\mathbb{B}$ = 10 and 4 respectively. For the specific combination of random seed and the sampling prior used here, this means that the winning Stack-LSTM saw maximum values of $n = 18, m = 20$ during training, and generalized to all strings up to $a^{120}b^{120}c^{240}$ with $M_{det} \geq 99.5\%$.

6.2 Perfect accuracy

We report the generalization scores using a strict $\varepsilon = 0$ as well, i.e., when a model is required to predict *all* deterministic steps correctly or assign non-zero probability to valid symbols only. For languages with deterministic steps such as $a^n b^n$, this means that the model needs to always predict the end-of-sequence symbol correctly, thus making a distinction between accepting a string and approximating its surface structure.

Here, only MDLRNNs remain at the same scores, indicating that they predicted all time steps correctly. Baby-NTM reaches $\mathbb{B} = 1$ for $a^n b^n$, a drop from 10. The rest of the networks drop to $\mathbb{B} < 1$, revealing that their good scores in the previous comparison calculated with a non-zero ε was due to them approximating the target languages, even at low n values.

MDLRNN performance here is in line with results from Lan et al. (2022), who provided evidence that MDLRNNs for these languages do not only perform empirically well on large test values, but are also provably correct for any input. However, here we limited activations to standard, non-discrete functions (Section 5.1), potentially limiting the network's ability to generalize well in the limit. While we do not provide correctness proofs for the networks found here, the index scores indicate that MDLRNNs generalize well to large values using only standard activations. Figure 4 presents the MDLRNN found for $a^n b^n$. We checked whether this network also accepts n values beyond those needed to reach the score $\mathbb{B} = 10$ (n = 10,020). The network reached 100% M_{det} for all values up to n = 35,000, at which point we stopped the test due to long feeding times.

Beyond the benchmark scores, Figure 3 plots the largest n value for $a^n b^n$ strings predicted by the models tested here with 100% M_{det} accuracy ($\varepsilon = 0$), as a function of training set size. Both MDLRNNs and Baby-NTMs reach perfect accuracy up to the tested maximum of n = 1,000. MDLRNNs however require two orders of magnitude less data to reach this performance (and the benchmark scores in Table 2 show that in fact MDLRNNs generalized up to at least n = 10,000, while Baby-NTMs remained at 1,000). LSTMs and Stack-RNNs did not generalize well beyond the training samples. This is in line with previous works showing that these models may need substantially more training data in order to learn these languages (Table 1).

7 Discussion

We provided a simple index for how well a model generalizes: how much it can learn from how little data. We illustrated the usefulness of this index in a comparison of several current models over several formal languages. Beyond showing which current models generalize better than others, the benchmark also highlights which aspects of artificial neural networks work well for grammar induction, and what is still missing.

Among languages that were learned with perfect accuracy (a^nb^n, $a^nb^mc^{n+m}$, Dyck-1), MDLRNNs generalized best, but still failed on others ($a^nb^nc^n$, $a^nb^nc^nd^n$, and Dyck-2). Previous work has shown that this model's search procedure, an evolutionary algorithm, fails to find networks that are manually shown to have better MDL scores (Lan et al., 2022). We take this to show that the optimization procedure limits the model and prevents it from taking full advantage of the MDL objective. The benefit of the MDL objective is nevertheless evident in the generalization performance for several languages.

MARNNs clearly benefit from their memory devices and reach good generalization scores, but testing for perfect accuracy ($\varepsilon = 0$) reveals that their learning outcome is mostly approximate, and that they fail to maintain perfect accuracy for long stretches beyond their training data. This could be the result of an inadequate objective function (cross-entropy), limitations of the search (backpropagation/gradient descent), or both. We do not currently have results that help decide this matter, but recent results for other architectures (El-Naggar et al., 2023b) hint that the problem lies at least in part in the objective function.

8 Acknowledgements

This work was granted access to the HPC resources of IDRIS under the allocation 2023-AD011013783 made by GENCI.

References

Mikael Bodén and Janet Wiles. 2000. Context-free and context-sensitive dynamics in recurrent neural networks. *Connection Science*, 12(3-4):197–210.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and Pedro A. Ortega. 2022. Neural Networks and the Chomsky Hierarchy.

Nadine El-Naggar, Pranava Madhyastha, and Tillman Weyde. 2022. Exploring the Long-Term Generalization of Counting Behavior in RNNs.

Nadine El-Naggar, Pranava Madhyastha, and Tillman Weyde. 2023a. Theoretical Conditions and Empirical Failure of Bracket Counting on Long Sequences with Linear Recurrent Networks.

Nadine El-Naggar, Andrew Ryzhikov, Laure Daviaud, Pranava Madhyastha, and Tillman Weyde. 2023b. Formal and empirical studies of counting behaviour in ReLU RNNs. In *Proceedings of 16th Edition of the International Conference on Grammatical Inference*, volume 217 of *Proceedings of Machine Learning Research*, pages 199–222. PMLR.

W. Tecumseh Fitch and Marc D. Hauser. 2004. Computational constraints on syntactic processing in a nonhuman primate. *Science*, 303(5656):377–380.

Felix Gers and Jürgen Schmidhuber. 2001. LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks*, 12(6):1333–1340.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural Turing Machines. *arXiv:1410.5401 [cs]*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Alon Jacovi, Avi Caciularu, Omer Goldman, and Yoav Goldberg. 2023. Stop Uploading Test Data in Plain Text: Practical Strategies for Mitigating Data Contamination by Evaluation Benchmarks.

Armand Joulin and Tomas Mikolov. 2015. Inferring Algorithmic Patterns with Stack-Augmented Recurrent Nets. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization.

Nur Lan, Michal Geyer, Emmanuel Chemla, and Roni Katzir. 2022. Minimum Description Length Recurrent Neural Networks. *Transactions of the Association for Computational Linguistics*, 10:785–799.

Raphaëlle Malassis, Stanislas Dehaene, and Joël Fagot. 2020. Baboons (Papio papio) Process a Context-Free but Not a Context-Sensitive Grammar. *Scientific Reports*, 10(1):7381.

William Merrill. 2021. On the Linguistic Capacity of Real-Time Counter Automata.

J. Rissanen. 1978. Modeling by shortest data description. *Automatica*, 14(5):465–471.

Hava T. Siegelmann and Eduardo D. Sontag. 1992. On the computational power of neural nets. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 440–449, New York, NY, USA. Association for Computing Machinery.

Mirac Suzgun, Yonatan Belinkov, Stuart Shieber, and Sebastian Gehrmann. 2019a. LSTM Networks Can Perform Dynamic Counting. In *Proceedings of the Workshop on Deep Learning and Formal Languages: Building Bridges*, pages 44–54, Florence. Association for Computational Linguistics.

Mirac Suzgun, Sebastian Gehrmann, Yonatan Belinkov, and Stuart M. Shieber. 2019b. Memory-Augmented Recurrent Neural Networks Can Learn Generalized Dyck Languages. *arXiv:1911.03329 [cs]*.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. On the Practical Computational Power of Finite Precision RNNs for Language Recognition. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 740–745.

A Appendix: Hyper-parameters

A.1 Training corpora

All training sets were generated using the same random seed 100 and prior probability $p = 0.3$. The datasets are available at https://github.com/taucompling/bliss. Following Jacovi et al. (2023), the datasets are zipped and password-protected to prevent test data contamination of large language models through crawling.

Each of the LSTM and MARNN hyper-param configurations below was run 3 times using different random seeds (100, 101, 102). MDLRNNs were run once per configuration because of their longer running time.

A.2 LSTM

LSTMs were trained based on the following hyper-params grid: hidden state size (2/32/128), regularization technique (L1/L2/none), and the regularization constant in case regularization was applied (λ = 1.0/0.1/0.01). Networks were trained using the Adam optimizer (Kingma and Ba, 2017) with learning rate 0.001, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The networks were trained by feeding the full batch of training data for 1,000 epochs.

A.3 MARNN

MARNNs were trained by varying the architecture type (Softmax Stack-LSTM/Softmax Baby-NTM) and stack/memory size (50/100 for Stack-LSTM, 2050 for Baby-NTM). For Stack-LSTM, stack sizes were selected so they were always larger than the largest values seen during training: $n + m = 22 + 24$ for $a^n b^m c^{n+m}$ and $n = 24$ for all other languages. During testing the stack size was enlarged to 2050, beyond the maximum needed to reach scores $\mathbb{B} = 1$ and 2. Baby-NTM memory was set to 2050 already during training because this model's memory size affects the weight dimensions and cannot be changed after training.

The rest of the hyper-parameters were set to the default values from Suzgun et al. (2019b). Stack-LSTM: hidden size 8; 1 layer; memory dimension 5; epochs 3/50; learning rate 0.01; Baby-NTM: hidden size 8; 1 layer; memory dimension 5; epochs 3/50; learning rate 0.01.

The original MARNN setup was modified here so that the network outputs represent probability distributions and not multi-label outputs. This was done by replacing the sigmoid outputs with a softmax layer and the MSE loss with cross-entropy.

A.4 MDLRNN

MDLRNNs were trained using the evolutionary algorithm and the same hyper-params reported in Lan et al. (2022): population size 500; islands size 250; 25,000 generations; tournament size 2; early stop after 2 hours of no improvement; elite ratio 0.001; migration interval 1,000 generations/30 minutes.

B Appendix: PCFGs

B.1 $a^n b^n$

$$S \rightarrow \begin{cases} aSb & 1-p \\ \varepsilon & p \end{cases}$$

B.2 $a^n b^m c^{n+m}$

$$S \rightarrow \begin{cases} aSc & 1-p \\ X & p \end{cases}$$

$$X \rightarrow \begin{cases} bXc & 1-p \\ \varepsilon & p \end{cases}$$

B.3 Dyck-1

$$S \rightarrow \begin{cases} (\ S\ )\ S & p \\ \varepsilon & 1-p \end{cases}$$

B.4 Dyck-2

$$S \rightarrow \begin{cases} (\ S\ )\ S & p/2 \\ [\ S\ ]\ S & p/2 \\ \varepsilon & 1-p \end{cases}$$

A Sanskrit grammar-based approach to identify and address gaps in Google Translate's Sanskrit-English zero-shot NMT

Amit Rao[1], Kanchi Gopinath
Plaksha University, Mohali, Punjab 140306, India
amitrao.human@gmail.com, gopinath.kanchi@plaksha.edu.in

Abstract

In this work, we test the working of Google Translate's recently introduced Sanskrit-English translation system using a relatively small set of probe test cases designed to focus on those areas that we expect, based on a knowledge of Sanskrit and English grammar, to pose a challenge for translation between Sanskrit and English. We summarize the findings that point to significant gaps in the current Zero-Shot Neural Multilingual Translation (Zero-Shot NMT) approach to Sanskrit-English translation. We then suggest an approach based on Sanskrit grammar to create a differential parallel corpus as a corrective training data to address such gaps. This approach should also generalize to other pairs of languages that have low availability of learning resources, but a good grammar theory.

1 Introduction and motivation

Translation between Sanskrit and English presents significant challenges, even for expert human translation, due to the unique features of Sanskrit and the large linguistic gap between Sanskrit and English. Also, Sanskrit has a unique role, specially in the Indian subcontinent as it was formerly a common language of communication across all fields. It was displaced by colonization, but is seeing a resurgence of late, specially for accessing Indian Knowledge Systems in the original. This has important implications on the expectations from an automatic translation, and the implications of erroneous translations.

In addition to the above factors, automatic machine translation between Sanskrit and English presents the additional challenge of Sanskrit having relatively low availability of high-quality training resources such as tagged corpora.

In May 2022, it was announced (Google, 2022) that 24 new languages including Sanskrit were being added to Google Translate, using the new Zero-Shot Machine Translation adaptation (Johnson et al, 2017) of Google's Neural Multilingual Translation (Wu et al, 2016).

In essence, Zero-Shot NMT leverages deep learning in a single model trained on multiple language pairs, to translate even between directions and language pairs it has not been explicitly trained on. Google's Zero-shot NMT is a variation of Zero-Resource Machine Translation (Firat et al, 2016), which requires an additional fine-tuning step using "pseudo-parallel" data of the new language pair. The need for this step is avoided in the design of zero-shot NMT.

It is interesting that Google has used the Zero-Shot NMT approach for Sanskrit-English. This is a data-driven approach to MT and NLP that avoids the need for explicit encoding of knowledge. The other possible approach to MT and NLP is grammar-based or model-driven, which needs explicit encoding of knowledge. The linguistic theory and grammar behind Sanskrit is very stable and sound. Using this theory base, efforts have been made to create grammar-based Sanskrit NLP systems, for example (Kulkarni, 2021).

The data-driven approach is normally attractive when training resource availability is good, and a good grammar model is absent or too complex. The

[1] Work carried out while the author was at Plaksha.

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 141–166
September 11–12, 2023. ©2023 Association for Computational Linguistics

grammar-based approach is normally attractive when there is a good grammar model, and the training resource availability is poor. With Sanskrit, there is a relatively good grammar model and relatively poor training resource availability. And yet, Google has chosen the data-driven approach for Sanskrit-English translation through its choice of zero-shot NMT, in order to maintain a uniform approach across all languages.

It is pertinent, therefore, to test the effectiveness of Google's Sanskrit-English translation system in actual use. We now describe the considerations we used to design this test.

2 Test Design Considerations for Sanskrit-English Google Translate

We now describe the considerations that came up when designing a test for Sanskrit-English Google translate, and how we dealt with them:

a. **Directionality** – Should we test Sanskrit to English, or English to Sanskrit or both? For the Sanskrit-English pair, we expect three main use cases:
   A. **Sanskrit-English for Sanskrit access** - People fluent in English and interested in Sanskrit literature trying to translate a traditional Vedic or classical Sanskrit mantra, shloka, poem or text from Sanskrit to English.
   B. **English-Sanskrit for Sanskrit learning or communication** - People fluent in English, interested in learning Sanskrit (either for conversation or to access the literature) and trying to translate English to Sanskrit.
   C. **Sanskrit-English for communication or learning** - People fluent in Sanskrit but not English, translating their original Sanskrit text into English for communicating with others or to learn English.

On account of the nature of the language pair and their current status, of the above three, we expect case A to overwhelmingly dominate, case B to be next, and case C to be relatively insignificant. Thus, the Sanskrit-English direction is the highest priority, and also has higher demands on accuracy, since it is more likely to be dealing with classical texts whose incorrect interpretation could have undesirable cultural consequences. We therefore put more emphasis on testing Sanskrit-English than English-Sanskrit. Most of our discussion will also be about Sanskrit to English translation, unless otherwise mentioned or obvious from the context.

b. **Purpose** - Our purpose is to check whether Google's zero-shot NMT automatic translation is robust enough for the Sanskrit-English language pair, rather than an exhaustive performance analysis of the translation. Therefore, rather than a test suite aimed at complete coverage of the language pair, we will create a probe test suite of a few carefully hand-crafted cases, leveraging grammar and language theory, mainly focused on areas where we expect challenges for translation.

c. **Automation** - Since the test set was small and for one-off use, and to be hand-crafted leveraging human expertise, it was simpler to do it iteratively and manually for now rather than invest effort in automating it.

d. **Sourcing** - We did not find a readily available translation test suite for Sanskrit-English focused on testing the robustness of zero-shot NMT, so we created our own.

3 Test process and results

Based on the considerations discussed above, a small "probe" test suite of approximately 120 test cases was hand-crafted and applied iteratively.

Sanskrit to English translation	English to Sanskrit translation
98 test cases	31 test cases

Table 1: Number of test cases

The test cases are not all independent, many are part of a group of inputs iteratively designed to test different variants of a specific area being tested. For example, correct translation of single/dual/plural number involves inputs containing various combinations of these. It is difficult to enumerate the groups, since there are sometimes overlaps where a single test case is logically part of multiple groups. Hence, the above table lists the number of individual test cases and not groups.

Each test case was manually translated by one of the authors, who is fluent in both Sanskrit and English, to create the expected reference output. The test case was then input to Google Translate and the output recorded against it. The output was manually evaluated using a 3-way rating system defined by us as follows.

Rating	Meaning
Ok	**Correct** - Output either matches reference output exactly, or is close enough and there is no change in meaning.
?	**Dubious -** Output is acceptable, but not ideal, and/or translation is not consistent across the group.
*	**Incorrect** - Output is totally unacceptable, as it conveys a totally unintended meaning.

Table 2: Rating system

All the test cases, the expected and actual outputs and ratings are available in the sheet attached as Appendix B – All Test Cases. The ratings are annotated with explanations in a Remarks column where needed.

The results of the test are summarized below.

Sanskrit to English translation		
Rating	**Count**	**Percent**
Ok	37	37.76
?	34	34.69
*	27	27.55
Total:	**98**	
English to Sanskrit translation		
Rating	**Count**	**Percent**
Ok	10	32.26
?	10	23.36
*	11	35.48
Total:	**31**	

Table 3: Test results summary

As seen from the above table, if we take the stricter criterion of only Ok-rated outputs as correct, the accuracy of Google Translate for our probe test is 37.76 per cent for Sanskrit to English, and 32.26 per cent for English to Sanskrit. If we take the more relaxed criterion of only *-rated outputs as incorrect, the accuracy is 72.45 per cent for Sanskrit to English, and 64.52 per cent for English to Sanskrit. Since this is only a probe test, and not an exhaustive coverage test, we cannot claim these as the actual accuracy figures, but the test probe does reveal that there are significant gaps in the performance of Google Translate for both the directions that need to be fixed before the translation can be considered robust. The detailed remarks about each output can be found in Appendix B – All Test Cases. In the following section, we summarize the key observations and their implications.

It must be noted here that Google Translate by design is a learning product and is therefore being continuously updated. The test results are valid as of the time they were conducted, namely, in the third week of December 2022.

4 Key observations and implications

Looking at the test case outputs, we find that given the multiple inherent challenges of Sanskrit-English translation, the system performs surprisingly well for a zero-shot NMT that has possibly not been trained on a single input specific to the Sanskrit-English pair. Of the specific challenge areas tested by the probe test, it does cover quite a wide spectrum of phenomena satisfactorily, in at least a few cases, including *sandhi, samāsa, taddhita,* dual number, three grammatical genders, and the phenomenon of *sati-saptamī* (absolutive locative clause).

Having said that, the output is often inconsistent across variations of a language feature, and dubious or incorrect in several cases.

For instance:

- It fails to disambiguate the word भवति[*bhavati*], based on the context, as the sambodhana (vocative) form of भवती[*bhavatī* - "lady"] rather than the third-person present tense form of the धातु[dhatu](verbal root) भू[*bhū*](to be/become).
- It fails to disambiguate the word नेत्रे[*netre*], based on the context, as the accusative case dual number form of the neuter gender noun नेत्र[netra](eye) rather than its locative case singular number form.

- It incorrectly uses Hindi/Urdu words such as कुर्सी, मेज़, and मौसम, which are totally absent in Sanskrit, rather than the corresponding Sanskrit words, to translate English words such as "chair", "table" and "weather" respectively.
- It fails to split the sandhi correctly in अहं ब्रह्मास्मि[*ahaṃ brahmāsmi*] based on the context, which causes it to interpret it as ब्रह्मा[*brahmā*](Brahma the creator) rather than ब्रह्म[*brahma*](Brahman, the Supreme Self), though it does recognize the distinction correctly in non-sandhi cases.

The above examples have been described and discussed in Appendix A – Example Test Cases.

Based on all the test cases listed in Appendix B, we summarize our analysis of the results of the probe test as follows:

A. Google Translate's zero-shot NMT for the Sanskrit-English language pair covers a fairly broad spectrum of translation phenomena. However, our probe test has revealed several significant gap areas that need to be addressed before the system can be considered robust and reliable for general use.

B. The gaps we have identified are largely a consequence of two system factors falling out of the zero-shot NMT data-driven approach –

   a. Not leveraging linguistic knowledge explicitly, due to the design decision of NMT (Wu et al, 2016)

   b. Not training with language-pair specific data, due to zero-shot usage to deal with low availability of resources (Johnson et al, 2017)

   For example, the training data may have included only translation data for Sanskrit-Hindi/Urdu and Hindi/Urdu-English. As a consequence of these, the system makes the following main types of errors: (a) Inconsistent translation across variants of the same phenomenon (b) The system sometimes erroneously translates into Hindi/Urdu words that do not exist in Sanskrit. (c) Unrecognized Sanskrit words are translated to the nearest similar sounding word seen in the training, which leads to errors.

C. These systemic gaps can be addressed by leveraging grammar and language theory. In particular, Sanskrit has an extremely well-developed grammar and language model that allows for precise and accurate representation of the meaning of a sentence.

5 Characterizing the gaps

The current gaps in Google Translate's English-Sanskrit translation, summarized in the previous section, can be classified into two categories:

1. **Learning gaps** - These are gaps that can be addressed by better training of the zero-shot NMT, by feeding more training data, or tuning the learning parameters. For example, if a specific English idiom is not currently learnt as an idiom, it could be learnt by feeding in examples of its usage in the English-Hindi/Urdu translation corpus. We can expect such gaps to gradually reduce over time as the system is fed with more training data, without any change to the basic zero-shot NMT approach. However, with the approach being suggested in this work, this gap reduction could be speeded up.
2. **Systemic gaps** due to pure zero-shot NMT - These are gaps that arise due to not feeding translation data specific to the target language pair (in our case English-Sanskrit) in training the system, but only leveraging translation data of other language pairs that between them cover the target language pair, for example, in our case, English-Hindi/Urdu and Sanskrit-Hindi/Urdu. Such gaps are inherent to the pure zero-shot NMT approach and will not reduce over time. Addressing such gaps needs a different approach that we shall touch upon shortly.

Let us try to formally characterize these two types of gaps in order to understand them better. In order to do that, we must first characterize different types of machine translation systems and see where zero-shot NMT fits in.

Essentially, a deep-learning based machine translation system is a transformer that takes a text *s* in the source language S and transforms it into a text *t* in the target language T. In order to do this, it uses a pre-trained language model.

Let **D(L1, L2)** represent training data consisting of parallel translations from the language L1 to the language L2.

Google Translate's Neural Multilingual Translation (NMT) approach (Wu et al, 2016) uses a single Large Language Model (LLM) that is trained on all the languages available. This allows learning to be leveraged across different data sets and languages. Thus, for example, the learning from D(L1, L2), D(L2, L3) and D(L4, L2) is merged into a single model. Then, in translating (L1, L2), the learning from not just D(L1, L2), but also D(L2, L3) and D(L4, L2) gets leveraged, leading to more robust output than from just D(L1, L2) alone. Moreover, this allows the system to translate even between language pairs even though that specific pair was not part of the learning data, say, due to low availability of training data for that pair. In this case, for example, the system could give a translation for (L1, L4), though this pair was not part of the training data. This would obviously not be as robust as having D(L1, L4) in the training mix, but may be better than giving no translation at all. This is what Google means by zero-shot NMT (Johnson et al, 2017), and that is what is reportedly used in English-Sanskrit translation. The implication is that the system is currently not trained on D(English, Sanskrit) data at all. It leverages, for example, D(English, Hindi/Urdu) and D(Sanskrit, Hindi/Urdu), and possibly data in other language pairs, to attempt (English, Sanskrit) translation.

Let **F(S,T)** be the set of features that would have been learnt by the NMT if it had been trained with the ideal training data set D(S,T) to correctly perform an arbitrary (S,T) translation request. Now, in zero-shot NMT, there is no D(S,T). Instead, the NMT is fed an *n*-member set D_n of $D(L_i, L_j)$ where $1 <= i <= n$, $1 <= j <= n$, and (L_i, L_j) is not in D_n. The assumption here is that F(S,T) will be compositionally learnt by the NMT via some combination of $D(L_i, L_j)$ training inputs.

We can now characterize the two categories of gaps we mentioned above in these terms.

A **learning gap** is one where F(S,T) is not currently achieved, but can be learnt by either adding more data to Dn, or by optimizing the parameters of the NMT, or both. For example, let us say a specific form of a verb in Sanskrit is not being correctly translated into English. This could be addressed by adding data that contains that form in the D(Sanskrit, Hindi/Urdu) set.

A **systemic gap** is one where no combination of $D(L_i, L_j)$ comprising D_n for any value of *n* can cause learning of F(S,T), because there exists a set of features **Fo(S,T)** that are not compositional, but can only be learnt by training on D(S,T).

Let us look at an example each of both these gaps.

First, an example of a learning gap. The word मां[*māṃ*] occurs both in Sanskrit and Hindi/Urdu. In Sanskrit, it is the sandhi form of माम्[*mām*] and means "me". In Hindi/Urdu, it is the simplified form of माँ[māṁ] and means "mother". Currently, Google Translate sometimes confuses these two cases, as seen from some of the test outputs. This distinction can be trained into the system by having more examples distinguishing the two cases in the D(Sanskrit, Hindi/Urdu) and D(Hindi/Urdu, English) training data. Hence, we can call this a learning gap.

Now, an example of systemic gap. The unambiguous Sanskrit sentence "अश्वे उपविशन् पिता पुत्रं पश्यति"[*aśve upaviśan pitā putraṃ paśyati*] is ideally translated into English as "The father seated on a horse sees his son". Similarly the unambiguous Sanskrit sentence "पिता अश्वे उपविशन्तं पुत्रं पश्यति"[*pitā aśve upaviśantaṃ putraṃ paśyati*] is ideally translated into English as "The father sees his son **who is** seated on a horse". In Sanskrit, the distinction between the two is clear and marked by inflection on the appropriate noun. In English, the distinction is clear when marked with a relative clause marked by "who is". However, a possible Hindi translation of both these would be "पिता अपने बेटे को घोड़े पर बैठे हुए देखता है"[*pitā apane beṭe ko ghoḍe para baiṭhe hue dekhatā hai*]. This literally translates to "The father sees his son seated on a horse", which is ambiguous due to the prepositional phrase attachment ambiguity, and can convey both the meanings. The issue here is that the unambiguous case markings of Sanskrit on the phrase "seated on the horse" get mapped in both the Hindi translations to a single oblique case marking (बैठे हुए[*baiṭhe hue*]), which has the effect of saying "seated on a horse" without adding the relative clause marker "who is". This feature mismatch (divergence pattern) between the languages (namely inflection in Sanskrit vs oblique case in Hindi vs relative clause in English) causes a non-compositionality in translation. Therefore there is an information loss in zero-shot NMT with transfer learning involving only D(Sanskrit, Hindi/Urdu) and D(Hindi/Urdu, English). This gap cannot be addressed by adding any amount of D(Sanskrit, Hindi/Urdu) and D(Hindi/Urdu, English) data or tweaking the parameters of the zero-shot NMT. It can only be addressed by adding examples of both Sanskrit sentences translated correctly directly to English, that is, by adding some D(Sanskrit, English) data. Thus, we can call this a systemic gap. [Note: This example is slightly simplified for ease of understanding. The actual Sanskrit-Hindi and Sanskrit-English translations by Google Translate are marginally different, but close enough for the example and the argument to hold. The actual details are discussed in Appendix C – "The Systemic Gap Example"].

Systemic gaps are an outcome of "language divergence" in translation, which was formally described in (Dorr, 1994). A partial set of language divergence patterns between English and Sanskrit was described in the context of a prototype rule-based machine translation for English to Sanskrit in (Mishra and Mishra, 2009).

The conclusion we can draw is that due to the presence of systemic gaps, the purely data-driven approach of zero-shot NMT, without encoding grammar knowledge, and without training on the specific language pair, does not work well enough for the Sanskrit-English language pair.

The zero-resource approach of (Firat et al, 2016) using "pseudo-parallel" Sanskrit-English data will not work in the presence of systemic gaps. Google's zero-shot NMT paper (Johnson et al, 2017) discusses (in section 4.6, "Zero-Shot Translation" and section 4.7, "Effect of Direct Parallel Data") possible approaches to go beyond zero-shot NMT by adding real parallel data in the missing language pair and direction (e.g. Sanskrit-English in our case). If a lot of high-quality real-life parallel data is available, this is found to be ideal. However, we know that high-quality Sanskrit-English real-life data is not readily available. For example, Samanantar, a parallel corpora collection between English and 11 Indic languages, does not include Sanskrit (Ramesh et al, 2022). A more recent effort, IndicTrans2, includes parallel corpora with English and all 22 scheduled Indian languages including Sanskrit (AI4Bharat et al, 2023). This is a good start, however, the size of the corpus is only of the order of 2.5k real-life sentences, which may be insufficient for good quality learning. Also, the direction is English-Sanskrit, and reversing it for Sanskrit-English training would not be ideal.

To address this need for direct parallel Sanskrit-English data to plug the gaps in zero-shot NMT, and the absence of such large high-quality corpora, we suggest the following approach – construct a "differential corpus" as a corrective data suite of Sanskrit-English language-pair-specific training data, by leveraging such linguistic knowledge. After all, we leveraged linguistic knowledge to create an effective probe test without requiring an exhaustive test suite. In the same way, we propose that this linguistic knowledge could be leveraged to create the right training data to address these areas and plug the gaps to create a more robust zero-shot NMT system.

To our knowledge, there is no readily available framework that can be directly leveraged to create such a differential corpus. The work on language divergence in general (Dorr, 1994) and English-Sanskrit (Mishra and Mishra, 2009) in particular cited earlier is a good directional starting point, but we will need to focus it on the Sanskrit-English direction and cover a more comprehensive set of features specific to the systemic gap issue than addressed in those works. There has been work done in identifying the formal structure of Sanskrit text (Huet, 2009) that could provide a set of features for us to take into account, but they may need to be filtered to keep only the ones relevant from the viewpoint of divergence. There is also a translation of the exercises from Apte's classical text on Sanskrit syntax (Apte, 1885) into Sanskrit (Gillon, 1996). This again may need to be filtered for divergence specific cases. A generic grammar framework like The Grammatical Framework (GF, 1998) is potentially interesting for its ability to deal with multilingual grammars, but it currently has support only for English and Hindi, not Sanskrit. Sanskrit-specific tools and toolkits such as Inria's Sanskrit Heritage Site (Huet, 1994), the University of Hyderabad Department of Sanskrit's Samsaadhani (Kulkarni, 2002) and Ashtadhyayi.com (Bodas, 2015) are targeted at understanding Sanskrit rather than translating it to English, so they may be useful as reference tools for the human experts generating the differential corpus.

Large language models (LLMs) like ChatGPT (Yiheng Liu et al, 2023) get their capabilities through pre-training, an expensive and long drawn-out process. Reinforcement Learning from Human Feedback (RLHF) (Ziegler et al, 2019) can be used to affect their behaviour (or to "fine-tune" it, such as eliciting right responses or suppressing objectionable output), which is not as expensive, and hence can be undertaken multiple times. RLHF can potentially be used with the differential corpus suggested, but this needs further study.

In-context learning (Sang et al, 2022) is seen as a major shift in transfer learning in the context of LLMs, with intimations of an "emergent" behaviour. In contrast to the classic pretraining-then-finetuning procedure for downstream prediction tasks in LLMs, there is only a need to provide a few "in-context" examples, without affecting existing model parameters. The differential corpus suggested might well function as in-context examples that can be taken up as future work.

The proposed differential corpus would consist of D(Sanskrit, English) and D(English, Sanskrit) data that would be designed, based on knowledge of Sanskrit and English grammar theory, to focus on addressing the systemic gap, that is, exercising the language features Fo(Sanskrit, English) and Fo(English, Sanskrit) that are not compositionally learnable from D(S,T) data sets not containing the above two language pairs.

In addition, since we are going to cover all combinations of a given feature, in the process, it may incidentally cover some learning gap data as well, because, intuitively, if a feature can be compositionally learnt from $D(S, L_1)$ and $D(L_2,T)$, then it can also be directly learnt from D(S,T). Thus, the suggested approach would also speed up reduction of the learning gap.

6 Outline of grammar-based approach for identifying a "differential corpus" as corrective training data

The key idea behind the proposed grammar-based approach is to leverage the rich linguistic model of Sanskrit from the traditional Indian *śāstra*s including vyākaraṇa (the aṣṭādhyāyī of Pāṇini and its related works), the vākyapadīyam of Bhartṛhari, mīmāṃsā, nyāya and vaiśeṣika, mapping approximately to linguistics, grammar, discourse analysis, logic and ontology respectively, to create a "differential corpus" of translation test cases that can be used as training data to fill the current gaps for Sanskrit-English zero-shot NMT.

The proposed approach is summarized as follows:

A. Identify the prominent divergence areas of the Sanskrit-English language pair, that is, the set of language features that are present in Sanskrit and either absent or rudimentary in English, Fo(Sanskrit, English) or vice versa, Fo(English, Sanskrit). For reasons stated earlier, we focus here on the first case, Fo(Sanskrit, English).

B. For each feature, iteratively create a group of test cases to test the translation of that feature. A group consists of a set of individual test cases. Collectively, the group should cover the range of variations of that feature. For example, if the divergence area is – presence of three grammatical numbers (singular/dual/plural) in Sanskrit, versus only two grammatical numbers (singular/plural) in English - the feature is "grammatical number", and we have to create as test input a group of sentences containing all combinations of singular, dual and plural nouns.

Of particular interest are cases of ambiguity, where two or more features map to the same form (e.g. a tiṅanta and subanta, or a kṛdanta and subanta, map to the same form as seen in the भवति example). The test inputs should check whether the translation deals with the ambiguity and provides the correct translation.

Such a differential corrective parallel corpus can be fed to the existing zero-shot NMT in addition to the training data it has already seen, without the need for any significant modification to the architecture of the system.

We now identify the key linguistic features of Sanskrit that are part of the proposed approach as outlined above, and highlight the potential challenge areas to be tested in each.

A. Lexical features

1. Sandhi - correct identification of all स्वर[svara](vowel), व्यञ्जन[vyañjana](consonant) and विसर्ग[visarga](aspirant) sandhis. Particularly where sandhi leads to ambiguous forms. For example: ब्रह्मास्मि[brahmāsmi] can be ब्रह्म

अस्मि[brahma asmi] or ब्रह्मा अस्मि[brahmā asmi].

2. **Special signs** - such as the अवग्रह-चिह्न[*avagraha-cihna*]ऽ.
   For example: अनुगृहितोऽस्मि[*anugṛhito'smi*] / अनुगृहितोस्मि[*anugṛhitosmi*] / अनुगृहितः अस्मि[*anugṛhitaḥ asmi*] are equivalent.

B. Morpho-syntactic features

3. सुबन्त[subanta](noun) forms - correct handling of ambiguous forms such as ते.

4. Basic तिङन्त[tiṅanta](verb) forms - correct handling of same धातु[dhātu] in multiple गणs[*gaṇa*s](groups)having same forms with different meanings, or having same form as subantas (e.g. भवति[bhavati]).

5. Derived tiṅanta(verb) forms - correct handling of verbs derived from णिच्[ṇic], सन्[san] and similar प्रत्ययs[pratyayas] (suffixes).

6. Compound sentences - correct handling of यद्/तद्[yad/tad] and similar conjoint sentences.

7. Complex sentences - correct handling of clauses involving कृदन्त[kṛdanta] (participials).

C. Semantic features

8. Word order and topicality - word order does not change the gross meaning in Sanskrit, but may alter the focus and topicality. Also, in some cases the order does matter, for example, placement of अपि[api] at the beginning vs middle.

9. तद्धित[taddhita] (noun-noun morphology) - for example, अण्[aṇ] patronymic pratyaya

10. समास[samāsa] (compound nouns) - correct translation of all the main samāsa types - तत्पुरुष[tatpuruṣa](including कर्मधारय[karmadhāraya], द्विगु[dvigu], उपपद[upapada], and नञ्[nañ]), बहुव्रीहि[bahuvrīhi], अव्ययीभाव[avyayibhava] and द्वन्द्व[dvandva]. For example – the same samāsa (e.g. पीताम्बर) can be interpreted as तत्पुरुष[tatpuruṣa] or बहुव्रीहि[bahuvrīhi] depending on the context.

A parallel corpus based on the above feature set could be created by leveraging the related work discussed earlier, as well as taking example sentences given for different grammar features of Sanskrit from a good Sanskrit grammar book, for example, (Rao, 2022), extrapolating them for complete coverage of all variations, and providing English translations. In some cases, the sentences may have to be hand-crafted as we have done here. Since we are looking at only a differential corpus, we estimate the number of test groups to be of the order of approximately a thousand in number, which is feasible to do manually in a reasonable time frame.

We believe a "differential" corrective translation data suite based on this model will allow most of the gaps in Google Translate's zero-shot NMT for Sanskrit-English to be addressed, leading to a more robust and usable system.

7 Contributions and scope for future work

This work has made the following contributions:

1. Through a small hand-crafted probe-test suite, we have shown that though Google Translate's recently introduced Sanskrit-English service based on zero-shot NMT covers a broad spectrum of cases adequately, there are still significant gaps in translation performance.
2. We have identified that the gaps are either learning gaps due to inadequate training data or need for parameter tuning, or systemic gaps due to the nature of zero-shot NMT and the divergence between the languages, and both these can be addressed by leveraging Sanskrit linguistic knowledge available in traditional works such as Bhartrhari's Vakyapadiyam (Sharma, 2016), and the *vyākaraṇa, mīmāṃsā* and *nyāya-vaiśeṣika* traditions that it references.
3. We have proposed an approach for creating corrective translation data for Sanskrit-English translation to address the systemic

gaps identified. We believe this idea is generalizable to other language pairs where there is a divergence in the language pair and a rich linguistic knowledge base exists.

Scope for future work includes:

1. Extending the approach to include English to Sanskrit direction considerations.

2. Creating an actual differential corrective translation test suite based on the approach.

3. Applying the differential corrective suite to Google Translate and measuring its impact.

Limitations

This work is subject to the following known limitation:

The solution proposed is currently indicative and directional, based on a theoretical understanding of how Google's zero-shot NMT works, and how the demonstrated gaps may have arisen, based on published literature on the system, and the authors' experience with translation and linguistics. The authors have not had access to the source code of the system, or any involvement with the actual training of the system. The proposed solution needs to be detailed out and practically implemented, ideally in collaboration with Google.

Ethics Statement

To the best of the authors' knowledge and belief, this work is fully compliant with the ACL Ethics Policy. We have identified significant gaps in the current working of zero-shot NMT for the Sanskrit-English pair, and have made suggestions for addressing these gaps. We believe these suggestions, if successfully implemented, will lead to a more robust and accurate system, thus improving the state of the art, which will benefit all stake-holders.

References

Google Blog. 2022. Google Translate learns 24 new languages. *Google IO announcement on Twitter*. https://twitter.com/Google/status/1524438946293374986.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi et al. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *Computing Research Repository*, arXiv1609:08144. Version 2.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat et al. 2017. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Computing Research Repository*, arXiv1611:04558. Version 2.

Mike Schuster, Melvin Johnson, and Nikhil Thorat. 2016. Zero-Shot Translation with Google's Multilingual Neural Machine Translation System. *Google Research Blog post.* https://ai.googleblog.com/2016/11/zero-shot-translation-with-googles.html.

Amba Kulkarni. 2021. Sanskrit Parsing Following Indian Theories of Verbal Cognition. In *ACM Transactions on Asian and Low-Resource Language Information Processing,* Volume 20, Issue 2, March 2021. Article No. 23, pages 1-38. https://dl.acm.org/doi/10.1145/3418061.

Raghunath Sharma. 2016 edition. वाक्यपदीयम् – Vakyapadiyam of Bhartrhari with Two Sanskrit Commentaries. Sampurnanand Sanskrit University, Varanasi, India. 2016 edition. ISBN 978-81-7270-020-1.

Tilaka Rao. अष्टाध्यायीप्रवेशः[*aṣṭādhyāyīpraveśaḥ*] vol.1 &2. 2022. Siddhanta Knowledge Foundation, Chennai, India. 2022 edition. ISBN 978-81-9569-142-5 and 978-81-9569-143-2.

Bonnie Dorr. 1994. Machine Translation Divergences: A Formal Description and Proposed Solution. *Computational Linguistics* 20(4):597-633.

Mishra, V. and Mishra, R. B. 2009. Divergence patterns between English and Sanskrit Machine Translation. *INFOCOMP Journal of Computer Science* 8(3): 62–71.

Huet, G. 2009. Formal Structure of Sanskrit Text: Requirements Analysis for a Mechanical Sanskrit Processor. In: Huet, G., Kulkarni, A., Scharf, P. (eds) *Sanskrit Computational Linguistics*. ISCLS ISCLS 2007-2008. Lecture Notes in Computer Science, volume 5402. Springer, Berlin, Heidelberg.

Pawan Goyal and Amba Kulkarni. 2014. Converting Phrase Structures to Dependency Structures in Sanskrit. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers:1834-1843, Dublin, Ireland.*

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.

Vaman Shivaram Apte. 1885. The Student's Guide to Sanskrit Composition. A Treatise on Sanskrit Syntax for Use of Schools and Colleges. Lokasamgraha Press. Pune, India.

Brendon S Gillon. 1996. Word Order in Classical Sanskrit. Indian Linguistics, 57(1-4):1-35.

Gowtham Ramesh, Sumanth Doddapaneni and 16 other authors. 2022. Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

AI4Bharat and 14 other authors. 2023. IndicTrans2: Towards High-Quality and Accessible Machine Translation Models for all 22 Scheduled Indian Languages. *arXiv preprint arXiv:2305.16307* (2023).

GF. 1998. Grammatical Framework. A programming language for multilingual grammar applications. https://www.grammaticalframework.org/.

Gerard Huet. 1994. The Sanskrit Heritage Site. https://sanskrit.inria.fr/index.fr.html .

Amba Kulkarni. 2002. संसाधनी[*saṃsādhanī*]. A Sanskrit Computational Toolkit. https://sanskrit.uohyd.ac.in/scl/ .

Neelesh Bodas. 2015. ashtadhyayi.com. A comprehensive study resource for the students of Panini grammar. https://ashtadhyayi.com/ .

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An Explanation of In-context Learning as Implicit Bayesian Inference. *arXiv:2111.02080 [cs.CL]* ICLR 2022 .Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, Geoffrey Irving. 2019. Fine-Tuning Language Models from Human Preferences. *arXiv:1909.08593* .

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, Zihao Wu, Dajiang Zhu, Xiang Li, Ning Qiang, Dingang Shen, Tianming Liu, Bao Ge. 2023. Summary of ChatGPT/GPT-4 Research and Perspective Towards the Future of Large Language Models. *arXiv:2304.01852 [cs.CL]* .

Appendix A – Example test cases

Example 1 (Ok):

Consider the following example of Sanskrit to English translation by the system. Though it was not a part of the probe test and therefore not in the appendix, it was the sentence that piqued our interest in carrying out this probe test.

Test input	**Reference (expected) output**	**Google Translate output**	**Rating and remark**
भ्रम्(१/४) धात्वोः विषये भ्रमः(४) मा भूत् इति व्याख्यानप्रपञ्चे वयं भ्रमन्तः(१) स्मः। [*bhram(1/4) dhātvoḥ viṣaye bhramaḥ(4) mā bhūt iti vyākhyānaprapañce vayaṃ bhramantaḥ(1) smaḥ*]	We are wandering(1) in the world of explanation so that there should be no confusion(4) about the verb bhram(1/4).	We are wandering(1) in the world of explanation so that there should be no confusion(4) about the verb bhram(1/4).	Ok – correctly disambiguates the two senses of bhram (wandering and confusion) based on the context.

Table 4: Example 1 (Ok)

This sentence is potentially challenging to translate,because the dhaatu(verbal root) भ्रम् [bhram] in Sanskrit occurs in two *gaṇa-s* (verb groups) (namely, 1 and 4), with different connotations, namely, "to wander" in group 1, and "to be confused" in group 4. Many of their forms are similar. Therefore, the sentence involves wordplay and ambiguity, which is traditionally a challenge for translation. However, Google Translate correctly translates these two senses. We could therefore say that the system has "learnt" the two senses of the verb root. However, with deep learning, such inputs are never explicitly encoded. Moreover, with zero-shot NMT, it is likely that the system was never fed an instance of Sanskrit to English translation of either of these senses. And yet, the system exhibits learning behaviour for this translation pair. This illustrates the power of deep learning and zero-shot NMT that causes learning without structuring and encoding of knowledge.

Along similar lines, the probe test analysis reveals that the zero-shot NMT has "learnt" a number of language phenomena that are potentially challenging for Sanskrit-English, in at least a few cases - *sandhi, samāsa, taddhita,* dual number, 3 grammatical genders, and the phenomenon of *sati-saptamī* (a kind of absolutive locative clause).

On the other hand, the system gives dubious, incorrect or inconsistent output for a number of cases, as seen in the remaining examples.

Example 2 (Error):

The word form भवति[*bhavati*] in Sanskrit is [ambiguous between the vocative form of the noun

Test input	Reference (expected) output	Google Translate output	Rating and remark
भवति भिक्षां देहि। [bhavati bhikṣāṃ dehi]	Madam, give me alms.	Give me the alms you have.	* - Does not recognize भवति[bhavati] as the sambodhana (vocative) of भवती[bhavatī - "lady"] .

Table 5: Example 2 (Error)

भवती[*bhavatī*](lady), and the simple present tense of the verb भू[*bhū*](to be/become). The verbal form is much more common in use than the vocative noun form. A grammar-based analysis would be able to deal with this ambiguity using knowledge-based disambiguation rules; a data-driven system would pick the statistically more common meaning, unless specifically exposed to this instance, which is highly unlikely with zero-shot NMT. This test case is a famous sentence from the famous epic Ramayana. Sentences from the epics are likely to be commonly queried for Google Translate, and getting it wrong is a fairly serious gap.

Example 3 (Dubious):

Test input	Reference (expected) output	Google Translate output	Rating and remark
नेत्रे पश्यतः। [netre paśyataḥ]	(Two) eyes see.	Looking into the eyes.	? - Confuses dual-number neuter-gender form with the *saptamī-vibhakti* (locative case) form.

Table 6: Example 3 (Dubious)

Similarly, the word form नेत्रे [*netre*] is ambiguous between dual number nominative/accusative case, or singular locative case. Here again, in the context of the word पश्यतः [*paśyataḥ*], the overall sentence is unambiguous in the light of Sanskrit grammar, however zero-shot NMT currently fails to get it right. This is not a classical sentence, but a simple Sanskrit sentence that is expected to be correctly translated.

There is another point this example serves to illustrate. In translation from language A to language B, if language A has a feature F that is absent in language B, and we are translating a sentence from A to B that involves the use of this feature F, then by default the canonical translation into B will lead to loss of information of the feature F. In this case, for example, since Sanskrit has the dual number while English does not, the sentence with the dual number would be folded to the plural number in English, thus leading to loss of information. One way to deal with this is to explicitly insert this information in some way in the target language, as we have done by adding "two" in parentheses in the reference expected output for this sentence. Whether to do this or not is a matter of choice, but the choice should be exercised uniformly for consistency. Examining the probe test cases in detail, we find that since zero-shot NMT does not explicitly deal with encoding any language feature such as number, the output is inconsistent, and depends on the training data instances that it has been exposed to.

Example 4 (Inconsistent):

Test input	Reference (expected) output	Google Translate output	Rating and remark
अहं ब्रह्म अस्मि। *[ahaṃ brahma asmi]*	I am the Brahman.	I am the Brahman.	Ok - Understands distinction of ब्रह्मन् (Brahman - neuter gender word) vs ब्रह्मा (Brahma - masculine gender word).
अहं ब्रह्मा अस्मि। *[ahaṃ brahmā asmi]*	I am Brahma.	I am Brahma.	Ok
अहं ब्रह्मास्मि। *[ahaṃ brahmāsmi]*	I am the Brahman.	I am Brahma.	? - Does not handle the ambiguity due to *dīrgha-sandhi* correctly, leading to incorrect output.

Table 7: Example 4 (Inconsistent)

अहं ब्रह्मास्मि [*ahaṃ brahmāsmi*] is an iconic sentence from the Upanishads and is considered a महावाक्य [*mahāvākya*] (great statement) of the Vedic literature, Sanatana Dharma and Hinduism.

It should translate to “I am Brahman” (the Ultimate Reality) and not “I am Brahma” (the four-headed Creator of the world, one of the Trinity of Brahma, Vishnu and Shiva). Google Translate is aware of this distinction, as seen from the first two examples, where the words are separated. However, it fails to recognize and translate it correctly when it is combined as a single word using sandhi. This is problematic and needs to be addressed.

Example 5 (Error):

Test input	Reference (expected) output	Google Translate output	Rating and remark
The chair is made of wood.	आसन्दं काष्ठेन निर्मितम्।	कुर्सी काष्ठेन निर्मिता भवति।	* - Uses Hindi/Urdu word कुर्सी for chair.
The table is made of wood.	उत्पीठिका काष्ठेन निर्मिता।	मेजः काष्ठेन निर्मितः अस्ति।	* - Uses Hindi/Urdu word मेज़ for table.
He did not come to work today as he is feeling a bit under the weather.	सः अद्य कार्यं कर्तुं न आगतः यतः सः किञ्चित् अस्वस्थः अस्ति।	सः अद्य कार्यं कर्तुं न आगतः यतः सः किञ्चित् मौसमस्य अधः अनुभवति।	* - Uses Hindi/Urdu word मौसम for weather. Also, does not understand the idiom "under the weather".

Table 8: Example 5 (Error)

The above three examples are from English to Sanskrit translation. In all three examples, English words have been translated using Hindi/Urdu words derived from Persian/Arabic and which are not Sanskrit words. This indicates that the zero-shot NMT was probably trained on Hindi/Urdu single-language data, and/or English-Hindi/Urdu sentence translation data, and this learning has percolated into English-Sanskrit translation. This is highly problematic for users who are trying to learn Sanskrit, as they will pick up words which are not in Sanskrit and assume them to be Sanskrit words.

Appendix B – All test cases

Part 1 of 2: Sanskrit to English

Test Input	Reference (expected) Output	GoogleTranslate Output	Rating	Remark
मम नाम अमितः।	My name is Amit.	My name is Amit.	Ok	Understands basic sentence with implicit copula "is".
अमितः मम नाम।	Amit is my name.	Amit is my name.	Ok	Seems to handle simple word order variation.
अहं मुम्बईतः।	I am from Mumbai.	I am from Mumbai.	Ok	Understands तः pratyaya used in place of fifth vibhakti.
पादोनसप्तवादनम्।	(It is) a quarter to seven.	It was seven o'clock in the morning.	*	Ignored पादोन for quarter-to. Where does morning come from?
पादोननववादनम्।	(It is) a quarter to nine.	Ninety-nine feet.	*	Confused पादोननव with ninety-nine, ignored वादनम् o'clock.
इदानीं पादोनसप्तवादनम्।	It is now a quarter to seven.	It is seven-foot-seven.	*	Ignored इदानीम् (for now), and confused पाद with feet.
पर्वतो वह्निमान् धूमात्।	The mountain is on fire because it has smoke.	The mountain is on fire from the smoke.	Ok	Understands sandhi.
मूकं करोति वाचालं पङ्गुं लङ्घयते गिरिम्। यत्कृपा तमहं वन्दे परमानन्दमाधवम्॥	He whose grace makes the mute speak and the lame cross mountains, I salute that supreme bliss Madhava.	He makes the speechless dumb and the lame cross the mountain I salute Him whose grace is the supreme bliss, Madhava.	?	Does not handle complex यद्/तद् properly?
या कुन्देन्दुतुषारहारधवला या शुभ्रवस्त्रावृता। या वीणावरदण्डमण्डितकरा या श्वेतपद्मासना॥ या ब्रह्माच्युतशङ्करप्रभृतिभिर्देवैस्सदावन्दिता। सा मा पातु सरस्वती भगवती निःशेषजाड्यापहा॥	She whose whiteness would defeat that of the kunda flower and the moon and of snow, who is adorned with white clothes, whose hands are adorned with the staff of a superior veena, who is seated on a white lotus, who is worshipped by gods beginning with Brahma, Achyuta and Sankara, may that goddess Saraswati, destroyer of all illusion, protect me.	She is white with necklaces of kunda and moon and snow and covered with white clothes That woman with her hands adorned with the staff of a veena and her seat on a white lotus She is always worshiped by the gods like Brahma, Acyuta and Shankara. May that holy Saraswati who destroys all illusion protect me	?	Largely ok, but does not handle complex yad/tad construct, does not get some complex samasas right, and does not understand that this is a single long sentence and needlessly splits it up.

अहं ब्रह्म अस्मि।	I am Brahman.	I am the Brahman.	Ok	Understands distinction of ब्रह्मन् (N - Brahman) vs ब्रह्मा (M - Brahma).
अहं ब्रह्मा अस्मि।	I am Brahma.	I am Brahma.	Ok	
अहं ब्रह्मास्मि।	I am Brahman.	I am Brahma.	?	Does not handle the ambiguity due to deergha sandhi correctly, leading to input that is actually incorrect.
रामकृष्णौ आगच्छतः।	Rama and Krishna are coming.	Ramakrishna and Rama are coming.	*	Does not handle द्वन्द्वसमास correctly?
मां तारय।	Save me.	Save me, mother.	?	Confused मां (माम्) with Hindi word माँ (mother).
मा तारय।	Do not save.	Don't save me.	?	Where did the "me" come from?
सा गजगामिनी।	She has a gait like an elephant.	She is an elephant.	*	Does not understand गजगामिनी as a samasa.
रामः पीताम्बरः।	Rama is one whose dress is yellow.	Rama is dressed in yellow.	Ok	Understands bahuvrihi vs tatpurusha use of पीताम्बर.
सः पीताम्बरः।	He is one whose dress is yellow.	He's in yellow.	?	Why the inconsistency "in yellow" vs "yellow-clad" across genders?
सा पीताम्बरा।	She is one whose dress is yellow.	She's yellow-clad.	?	Why the inconsistency "in yellow" vs "yellow-clad" across genders?
सः पीताम्बरं धारयति।	He is wearing a yellow dress.	He is wearing a yellow robe.	Ok	Understands bahuvrihi vs tatpurusha use of पीताम्बर.
भवति भिक्षां देहि।	Madam, give me alms.	Give me the alms you have.	*	Does not recognize भवति as sambodhana of भवती.
हे देवि, भिक्षां देहि।	Madam, give me alms.	O Goddess, give me alms,	Ok	
देवि भिक्षां देहि।	Madam, give me alms.	O Goddess, give me alms,	Ok	
रामः सीतया आगच्छति।	Rama is coming with Sita.	Rama is coming with Sita.	Ok	Understands all variations of

				upapada tritiya vibhakti with सह.
रामः सीतया सह आगच्छति।	Rama is coming with Sita.	Rama is coming with Sita.	Ok	
रामः सहसीता आगच्छति।	Rama is coming with Sita.	Rama is coming with Sita.	Ok	
रामः ससीता आगच्छति।	Rama is coming with Sita.	Rama is coming with Sita.	Ok	
वागर्थाविव सम्पृक्तौ वागर्थप्रतिपत्तये जगतः पितरौ वन्दे पार्वतीपरमेश्वरौ।	For attaining speech and its meaning, I salute the parents of the world, Parvati and Parameshwara, who are connected as speech and meaning.	I salute the two fathers of the world, the Supreme Lord of Parvati, who are as connected as the meaning of speech, for attaining the meaning of speech.		Does not understand वागर्थौ and पार्वतीपरमेश्वरौ as dvandva samasas. Does not understand पितरौ as an ekasesa samasa.
अनादिनिधनंब्रह्म शब्दतत्त्वंयदक्षरम्। विवर्तते अर्थभावेन प्रक्रिया जगतो यतः॥	The timeless eternal Brahman that is all-pervading, and whose transforms into meaning is the process of the world, is the essence of language.	The eternally immortal Brahman, the essence of sound, is the imperishable. Because the process of the world revolves around the sense of meaning	?	Does not understand the alternate meaning of अक्षरम् as pervasive. Does not properly understand yad/tad complex construct.
कर्मण्येवाधिकरस्ते मा फलेषु कदाचन। मा कर्मफलहेतुर्भूर्मा ते सङ्गोऽस्त्वकर्मणि॥	Your authority is over action alone, not over the fruits thereof, be not motivated by fruits of action, nor be attached to inaction.	You have only right to action, never to fruits. Let not the fruits of action be thy motive, nor let thy attachment be to inaction.	Ok	Understands this Bhagwadgita shloka.
अश्वत्थामो हतोहतः। नरो वा कुञ्जरो वा।	Ashvatthama is killed, whether man or elephant.	Ashvatthama was killed. Whether it is a man or an elephant.	Ok	Understands visarga sandhi.
अश्वत्थामो हतोहतः। नरोऽवा कुञ्जरोऽवा।	Ashvatthama is killed, whether man or elephant.	Ashvatthama was killed. A man or an elephant.	Ok	Understands avasarga, but why the change in translation?
अश्वत्थामो हतोहतः। नरोवा कुञ्जरोवा।	Ashvatthama is killed, whether man or elephant.	Ashvatthama was killed. Narova Kunjarova.	*	Does not split pada boundaries cleanly in absence of whitespace.
अश्वत्थामोहतोहतः। नरोवाकुञ्जरोवा।	Ashvatthama is killed, whether man or elephant.	Ashvatthamohatohata. Narovakunjarova.	*	Does not split pada boundaries cleanly in absence of whitespace.
ते गमिष्यन्ति।	They will go.	They will go.	Ok	
ताः गमिष्यन्ति।	They will go.	They will go away.	?	Where did "away" come from?

सः कुर्यात् सदा मङ्गलम्।	May he always do good.	May he always do good.	Ok	
सः क्रियात् सदा मङ्गलम्।	May he always do good.	He is always auspicious from action.	*	Confuses आशीर्लिङ् लकार form with (incorrect) panchami form.
ममोपात्त दुरितक्षयद्वारा श्रीपरमेश्वरप्रीत्यर्थम्।	For the pleasure of Sri Parameshwara through the destruction of evils attained by me.	For the pleasure of Sri Parameshwara through the destruction of evils attained by me.	Ok	Understands idiomatic usage like प्रीत्यर्थम्.
युगं वर्तते।	The age exists.	The age is present.	Ok	
युगे वर्तेते।	(Two) ages exist.	exists in the age.	?	Dual number not handled correctly and consistently.
युगानि वर्तन्ते।	Ages exist.	There are ages.	Ok	
युगम् अवर्तत।	The age occurred.	The era turned around.	?	Does not understand लङ्लकार (past tense) form अवर्तत of आत्मनेपद अकर्मक dhatu वृत्
युगे अवर्तेताम्।	(Two) ages existed.	Let them turn in the age.	*	
युगानि अवर्तन्त।	Ages existed.	The ages passed.	Ok	
नेत्रे पश्यतः।	(Two) eyes see.	Looking into the eyes.	?	Confuses dual number neutral gender form with saptami vibhakti form.
नेत्राभ्यां पश्यतः।	(They two) see with (two) eyes.	Looking at you with your eyes.	*	Where did "at you" and "your" come from?
श्रेयश्च प्रेयश्च मनुष्यमेतस्तौ सम्परीत्य विविनक्ति धीरः। श्रेयो हि धीरोऽभि प्रेयसो वृणीते प्रेयो मन्दो योगक्षेमाद्वृणीते ॥	The good and the pleasant both approach man. The wise, on examining both, chooses the good. The wise prefers the good over the pleasant, the unwise, compelled by material considerations, prefers the pleasant.	The steadfast man distinguishes between these two, the good and the dear. A sober person seeks the best of his dear ones, and a slow person seeks the safety of mystic yoga.	*	Does not understand meaning of योगक्षेम, does not analyze the shloka correctly.
सः नेत्राभ्यां पश्यति।	He sees through (two) eyes.	He looks through his eyes.	?	Not clear if it recognizes dual number here. Where did "his" come from?
अनुगृहीतोऽस्मि।	I am obliged.	I am gracious.	*	Does not understand standard phrase for "thank you" i.e. "I am obliged".

रामे वनं गते कृष्णः नगरं गतवान्।	When Rama went to the forest, Krishna went to the city.	When Rama went to the forest, Krishna went to the city.	Ok	Understands sati saptami.
रामे वनं गते सति कृष्णः नगरं गतवान्।	When Rama went to the forest, Krishna went to the city.	When Rama went to the forest, Krishna went to the city.	Ok	
रामः भोजनं कृत्वा शालां गच्छति।	Rama eats and goes to school.	Rama eats and goes to the shed.	?	Does not understand different meanings of शाला in context.
रामः अशित्वा शालां गतः।	Rama, having eaten, went to school.	Rama went to the shed without eating.	*	Confuses अशित्वा (having eaten) kridanta form, reverses the meaning.
रामः भोजनम् अशित्वा शालां गतः।	Rama, having eaten food, went to school.	Rama ate the food and went to the shed.	*	Seems to get the right meaning of अशित्वा here, but seems to be order-dependent. Also, misses out the nuance of "having eaten" by using "and".
रामः महामनाः।	Rama is (one who has) a great mind.	Rama is a great mind.	?	Does not understand bahuvrihi consistently. Rama (is one who) has a great mind.
रामः कम्बुग्रीवः।	Rama is (one who has) a neck shaped like a conch shell.	Rama is a conchshell.	*	Does not understand कम्बुग्रीव as bahuvrihi.
युधिष्ठिरः अजातशत्रुः।	Yudhishthira (is one who) has no enemies (born).	Yudhisthira is the unborn enemy.	?	Does not understand नञ् बहुव्रीहि samasa.
दाशरथिः रामः विजयते।	Rama, the son of Dasharatha, is victorious.	Rama, the charioteer of Dasaratha, is victorious.	*	Does not understand दाशरथि as taddhita form, son of Dasharatha. Seems to confuse with सारथी (charioteer).
सागरात् आहिमालयं भारतदेशः।	India (is) from the ocean upto the Himalayas.	From the sea to the Himalayas, India.	?	What about elided copula "is" which is mandatory in English?
आहिमालयं भारतदेशः।	India (is) upto the Himalayas.	The Himalayas are India.	*	Inconsistent recognition of

				avyayibhava samasa with आङ्
चिन्ता मास्तु।	Don't worry.	Don't worry.	Ok	
चिन्तामाऽस्तु।	Don't worry.	Let's worry.	*	Does not handle savarna deergha with avasarga.
रामस्य मनः महत्।	Rama's mind (is) great.	Rama's mind is great.	Ok	
रामस्य महामनः।	Rama's mind (is) great.	Rama's great mind.	?	Elided copula "is" not consistently inferred.
चैत्रवैशाखौ वसन्तऋतुः।	Chaitra and Vaishakha (are) the spring season.	Spring in Chaitra and Vaishakha.	?	Does not handle vidheya viseshana consistently well.
चैत्रवैशाखयोः वसन्तऋतुः।	The spring season (is) in Chaitra and Vaishakha.	Spring is the season of Chaitra and Vaishakha.	?	Confused shashti and saptami identical forms.
तस्य गतवैभवः पुनः न आयास्यति।	His lost glory will never return.	His lost glory will never come back.	Ok	
सः गतवैभवः।	He is one whose glory is gone.	He is a lost glory.	*	Does not understand bahuvrihi correctly.
तस्य प्राप्तविद्या महती।	His acquired knowledge is great.	His acquired knowledge is great.	Ok	
सः सम्प्राप्तविद्यः।	He is one who has properly acquired knowledge.	He is an acquired knowledge.	*	Does not understand bahuvrihi correctly.
कौन्तेयस्य अर्जुनस्य सारथिः श्रीकृष्णः।	Sri Krishna is the charioteer of Arjuna, the son of Kunti.	Sri Krishna is the charioteer of Kaunteya and Arjuna.	*	Does not handle taddhita and viseshana correctly.
गाङ्गेयः भीष्मः कौरवाणां सेनापतिः।	Bhishma, the song of Ganga (is) the commander of the army of the Kauravas.	Ganges, Bhishma, the commander of the army of the Kauravas.	*	Does not handle taddhita and viseshana correctly.
न जातु सः गृहं गच्छति।	He never goes home.	He never goes home.	Ok	Understands use of idioms like जातु.
जातु सः गृहं गच्छति।	Sometimes he goes home.	Jata he goes home.	*	But understanding is not consistent across usages.
प्राणवायवः पञ्चधा।	The prana-vayus are fivefold.	The prana-vayu is fivefold.	?	Understands धा pratyaya following N to mean Nfold, but not plural of वायवः?

रसाः षोढा।	The tastes are sixfold.	The juices are sixteen.	*	Does not understand non-standard form of धा pratyaya षोढा = sixfold and the sense of rasa as taste.
पदे द्विधा सुबन्तं तिङन्तं चेति।	There are two types of padams (words) - subanta and tinganta.	There are two types of verbs, the subjunctive and the triplet.	*	Does not understand vyakarana technical terms.
वनं पुलकितं सुन्दरम्।	The joyful forest is beautiful.	The forest is bright and beautiful.	?	Translation of पुलकित is inconsistent.
वनं पुलकितम्।	The forest is joyful.	The forest is thrilled.	?	Translation of पुलकित is inconsistent.
त्वं कुत्र असि?	Where are you?	Where are you?	Ok	
यूयं कुत्र स्थ?	Where are you (all)?	Where are you guys?	?	"Guys" seems too informal.
युवां कुत्र स्थः?	Where are you (two)?	Where are you guys?	?	Dual and plural information is not consistently differentiated.
भवान् कुत्र अस्ति?	Where are you?	Where are you at?	?	Where did "at" come from?
भवन्तौ कुत्र स्तः?	Where are you (two)?	Where are you two?	Ok	
भवन्तः कुत्र सन्ति?	Where are you (all)?	Where are you guys?	?	Dual and plural information is not consistently differentiated.
त्वं ग्रामे असि।	You are in the village.	You are in the village.	Ok	
युवां ग्रामे स्थः।	You (two) are in the village.	You are in the village.	?	Does not handle dual number and priority in case of mixed persons well.
यूयं ग्रामे स्थ।	You (all) are in the village.	You are in the village.	?	
भवान् ग्रामे अस्ति।	You are in the village.	You are in the village.	Ok	
भवन्तौ ग्रामे स्तः।	You (two) are in the village.	You two are in the village.	?	
भवन्तः ग्रामे सन्तिः।	You (all) are in the village.	You are in the village.	?	
गौः पर्णे खादति।	The cow eats (two) leaves.	The cow eats on the leaf.	?	Confuses neuter dual with saptami vibhakti. Cow eats on the leaf has no "yogyataa".

गौः द्वे पर्णे खादति।	The cow eats two leaves.	The cow eats two leaves.	Ok	Works ok if two is explicitly mentioned.
वृक्षात् पर्णे पततः।	(Two) leaves fall from the tree.	Leaves fall from the tree.	?	Does not distinguish dual from plural consistently.
वृक्षात् द्वे पर्णे पततः।	Two leaves fall from the tree.	Two leaves fall from the tree.	Ok	Works ok if two is explicitly mentioned.
यदि कयाचित् कवितया त्वधृदयम् अदारितं। तर्हि अहो बत! त्वया नैव काव्यम् अनुभूतम्॥	If your heart has not been torn apart by some poem, then oh my god! Poetry has not been experienced by you.	If any poem has torn your heart. Then oh my god! You have not experienced poetry.	*	Does not understand नञ् तत्पुरुष समास - अदारितम् = न दारितम्.
यदि कयाचित् कवितया त्वधृदयम् न दारितं। तर्हि अहो बत! त्वया नैव काव्यम् अनुभूतम्॥	If your heart has not been torn apart by some poem, then oh my god! Poetry has not been experienced by you.	If some poem has not torn your heart. Then oh my god! You have not experienced poetry.	Ok	

Part 2 of 2: English to Sanskrit

Test Input	Reference (expected) Output	GoogleTranslate Output	Rating	Remark
The chair is made of wood.	आसन्दं काष्ठेन निर्मितम्।	कुर्सी काष्ठेन निर्मिता भवति।	*	Uses Hindi/Urdu word कुर्सी for chair.
The table is made of wood.	उत्पीठिका काष्ठेन निर्मिता।	मेजः काष्ठेन निर्मितः अस्ति।	*	Uses Hindi/Urdu word मेज़ for table.
He did not come to work today as he is feeling a bit under the weather.	सः अद्य कार्यं कर्तुं न आगतः यतः सः किञ्चित् अस्वस्थः अस्ति।	सः अद्य कार्यं कर्तुं न आगतः यतः सः किञ्चित् मौसमस्य अधः अनुभवति।	*	Uses Hindi/Urdu word मौसम for weather. Does not understand the idiom "under the weather".
Armchair journalists on the internet are a dime a dozen.	आसन्दपत्रकाराः अन्तर्जाले बहवः।	अन्तर्जालस्य आर्मचेयरपत्रकाराः एकदर्जनं एकरूप्यकाणि भवन्ति।	*	Does not understand idioms such as "a dime a dozen".
Never were truer words spoken.	कदापि सत्यतरं वचनं न उक्तम्।	कदापि सत्यतरं वचनं न उक्तम् ।	Ok	
Truer words were never spoken.	सत्यतरं वचनं कदापि न उक्तम्।	सत्यतरं वचनं कदापि न उक्तम्।	Ok	
You are the apple of my eye.	त्वं मम नेत्रदीपः (महार्घः)।	त्वं मम नेत्रस्य सेबः असि।	*	Uses Hindi/Urdu word सेब for apple. Does not understand idiom "apple of the eye".
He has many servants at his beck and call.	तस्य सेवायै नैकाः सेवकाः सन्ति।	तस्य मुखे अनेके सेवकाः सन्ति।	?	Is मुखे the right translation of "beck and call"?
In order to develop an idea to its fullest, we must explore all facets of it.	कस्यचित् विचारस्य पूर्णविकासाय अस्माभिः तस्य सर्वाः पक्षाः अन्वेष्टव्याः।	विचारस्य पूर्णतया विकासाय अस्माभिः तस्य सर्वान् पक्षान् अन्वेष्टव्यम् ।	?	Should be सर्वे पक्षाः अन्वेष्टव्याः. Does not handle verb agreement for kartari/karmani correctly.
May the best team win.	उत्तमः दलः विजयं प्राप्नुयात्।	उत्तमं दलं विजयं प्राप्नुयात्।	?	दल in the sense of "team" is masc in Sanskrit
Thank you.	धन्यवादः।	धन्यवाद।	?	Should be धन्यवादः - Hindi influence?
Don't worry.	चिन्तामाऽस्तु।	चिंता मास्तु।	Ok	
How are you?	भवान् कथम् अस्ति?	भवान् कथमसि।	?	If भवान् is used, it should be अस्ति (प्रथमपुरुष)
I owe you.	अहं भवतः ऋणी अस्मि।	अहं भवतः ऋणी अस्मि।	Ok	Needs to assume male speaker, which is ok
She said, "I owe you".	"अहं भवत्याः ऋणिनी अस्मि" इति सा अवदत्।	सा अवदत्, "अहं भवतः ऋणी अस्मि" इति।	?	For female speaker being quoted, it should be ऋणिनी
She said that he was going home.	सः गृहं गच्छति इति सा अवदत्।	सा अवदत् यत् सः गृहं गच्छति।	Ok	
She said he was going home.	सः गृहं गच्छति इति सा अवदत्।	सा अवदत् यत् सः गृहं गच्छति।	Ok	
She said that he was going home yesterday.	सः ह्यः गृहं अगच्छत् इति सा अवदत्।	सा अवदत् यत् सः श्वः गृहं गच्छति।	*	श्वः is tomorrow, not yesterday! Seems to be Hindi influence, where the same word कल is used for both.

She said "I am going home".	"अहं गृहं गच्छामि" इति सा अवदत्।	सा अवदत् "अहं गृहं गच्छामि" इति।	Ok	
She said that she was going home.	अहं गृहं गच्छामि इति सा अवदत्।	सा गृहं गच्छामि इति अवदत्।	Ok	
She said she was going home.	अहं गृहं गच्छामि इति सा अवदत्।	सा गृहं गच्छति इति अवदत्।	?	Why the inconsistency with and without "that"?
Each time you open a book and read it a tree smiles knowing there is life after death.	यदायदापठनार्थं उद्घाटयसिपुस्तकम्। तदातदावृक्षस्स्मयते मरणोत्तरजीविज्ञः॥	प्रत्येकं पुस्तकं उद्घाट्य पठति तदा एकः वृक्षः मृत्योः अनन्तरं जीवनम् अस्ति इति ज्ञात्वा स्मितं करोति।	?	"Each time" should be "यदाकदाचित्"/"यदाकदापि" ("whenever")
Once the heart gets too heavy with pain, people don't cry. They just turn silent. They just turn silent completely.	वेदनया अभिभूतहृत्सु सत्सु नैव रुदन्ति जनाः। प्रत्युत निःशब्दा भवन्ति ते। केवलं पूर्णतया निःशब्दाः।	एकदा हृदयं वेदनायाः अतिभारं भवति तदा जनाः न रोदन्ति। ते केवलं मौनम् एव भवन्ति। ते केवलं सम्पूर्णतया मौनं कुर्वन्ति।	?	Here "once" should be "यदा" (when)
If a poem hasn't ripped apart your soul, you haven't experienced poetry.	यदि कयाचित् कवितया त्वधृदयम् न दारितं। तर्हि अहो बत! त्वया नैव काव्यम् अनुभूतम्॥	यदि कश्चन काव्यः भवतः आत्मानं न विदारितवान् तर्हि भवता काव्यस्य अनुभवः न कृतः।	Ok	
Ganesh and I are studying.	गणेशः च अहं च पठावः।	अहं च गणेशः पठामः।	*	Should be पठावः
I and Ganesh are studying.	अहं च गणेशः च पठावः।	अहं गणेशः च पठतः।	*	Should be पठावः
Harish, Ganesh and I are studying.	हरीशः च गणेशः च अहं च पठामः।	हरीशः, गणेशः, अहं च अध्ययनं कुर्मः।	Ok	
Harish, you and I are studying.	हरीशः च त्वं च अहं च पठामः।	हरीश, त्वं च अहं च अधीतवन्तौ।	?	Should be पठामः
You two and Ramesh are playing.	युवां च रमेशः च क्रीडथ।	भवन्तौ रमेशौ च क्रीडतः।	*	Should be क्रीडथ
Ramesh and you are playing.	रमेशः च त्वं च क्रीडथः।	रमेशः त्वं च क्रीडसि।	*	Should be क्रीडथः
Harish, Ramesh, you and I are playing.	हरीशः च रमेशः च त्वं च अहं च क्रीडामः।	हरीश रमेश त्वं च अहं च क्रीडतः।	*	Should be क्रीडामः. Does not handle mixed-person dual/plural cases correctly or consistently.

Appendix C – The Systemic Gap example

Sanskrit original	Ideal Hindi translation	Google Hindi translation
1. अश्वे उपविशन् पिता पुत्रं पश्यति [aśve upaviśan pitā putraṃ paśyati]	1(HI). घोडे पर बैठा पिता अपने बेटे को देखता है	1(HG). पिता अपने बेटे को घोड़े पर बैठते हुए देखता है
2. पिता अश्वे उपविशन्तं पुत्रं पश्यति [pitā aśve upaviśantaṃ putraṃ paśyati]	2(HI). पिता घोडे पर बैठे अपने बेटे को देखता है	2(HG). पिता अपने बेटे को घोड़े पर बैठा देखता है

Sanskrit original	Ideal English translation	Google English translation
1. अश्वे उपविशन् पिता पुत्रं पश्यति [aśve upaviśan pitā putraṃ paśyati]	1(EI). The father seated on the horse sees his son	1(EG). The father looks at his son as he sits on the horse
2. पिता अश्वे उपविशन्तं पुत्रं पश्यति [pitā aśve upaviśantaṃ putraṃ paśyati]	2(EI). The father sees his son who is seated on the horse	2(EG). The father sees his son sitting on the horse

The difference between Sanskrit sentences 1 and 2 is relationship of the phrase meaning “seated on a horse” with the father in sentence 1 and with the son in sentence 2. This is marked by inflection agreement with the appropriate nouns, so both sentences 1 and 2 in Sanskrit are clear and unambiguous irrespective of the word order.

In Hindi too, ideally the phrase meaning “seated on a horse” should be in agreement with the appropriate nouns (as shown in the ideal translations 1HI and 2HI). However, Google’s Hindi translations (1HG and 2HG) do not show this agreement. Instead, 1HG effectively means “The father sees his son sitting on a horse” and 2HG effectively means “The father sees his son seated on a horse”. Both these sentences use an oblique case marker to make the phrase “seated on a horse” a preposition phrase rather than an adjective phrase of the respective nouns. Such usage is increasingly common in everyday Hindi.

As a consequence of the choice of syntax, both sentences suffer from the same prepositional attachment problem that exists in their English meaning, and both become ambiguous and can represent sentence 1 as well as 2. In the main paper, for ease of presentation, we have combined 1HG and 2HG into a single Hindi sentence that represents the way both sentences 1 and 2 would be typically translated in Hindi.

From web to dialects: how to enhance non-standard Russian lects lemmatisation?

Ilia Afanasev
HSE University
Moscow, Russia
ilia.afanasev.1997@gmail.com

Olga Lyashevskaya
HSE University
Moscow, Russia
Vinogradov Russian Language Institute RAS
Moscow, Russia
olesar@yandex.ru

Abstract

The growing need for using small data distinguished by a set of distributional properties becomes all the more apparent in the era of large language models (LLM). In this paper, we show that for the lemmatisation of the web as corpora texts, heterogeneous social media texts, and dialect texts, the morphological tagging by a model trained on a small dataset with specific properties generally works better than the morphological tagging by a model trained on a large dataset. The material we use is Russian non-standard texts and interviews with dialect speakers. The sequence-to-sequence lemmatisation with the help of taggers trained on smaller linguistically aware datasets achieves the average results of 85 to 90 per cent. These results are consistently (but not always), by 1-2 per cent. higher than the results of lemmatisation with the help of the large-dataset-trained taggers. We analyse these results and outline the possible further research directions.

1 Introduction

Lemmatisation is a natural language processing (NLP) task that is a part of the basic language resource toolkit (BLARK) (Krauwer, 1998, 2003; Piotrowski, 2012). Lemmatisation may be defined as a transformation of a given token into the dictionary form, the latter being called a lemma. There may be different ways of lemmatisation, such as classifying a token by its particular supposed lemmatisation rule and the subsequent transformation by this rule (for instance, such model may classify *shown* into the group of tokens that are lemmatised with «delete last *n* and then add to before the token» rule, and then transformed by this rule into *to show*) (Anastasyev, 2020). In this paper, we focus on the sequence-to-sequence approach, which takes input sequence and transforms it into output sequence directly (Sutskever et al., 2014; Cho et al., 2014).

Sequence-to-sequence approach generally requires additional information for the token, because it is difficult for the model to lemmatise bare tokens (Kanerva et al., 2021). Many smaller lects[1] do not possess gold morphological tagging. However, they are located nearby a closely-related high-resource lect, for which there are a lot of gold morphological datasets.

We hypothesise that there is a reliable way to find a dataset with the specific distributional properties, train a tagger on it, use this tagger on a new, rather different dataset, and then lemmatise the tokens of this dataset with a preliminary fine-tuned large language model. We also presume that this approach is preferable to gathering the biggest data amount possible.

We believe that the thoroughness in morphological training data selection becomes gradually more important with increasing variation within the lemmatisation evaluation data. So, if overall the better tactic is to get the largest and the most heterogeneous dataset possible, for some types of data one needs a more nuanced approach.

We are going to demonstrate this on the material of the non-standard Russian lects. This includes the web as corpora material, social media texts, and dialect texts, presenting the continuum of lects getting further away from the standard Russian in terms of distributional properties. We hypothesise the following:

H1: Morphological tagging efficiency directly influences the lemmatisation accuracy.

H2: If the model trains on the larger dataset, the morphological tagging it performs will present stable satisfactory results.

H3: For the non-standard data, the distributional properties of the training dataset are generally more important than the sheer size.

We impose a set of restrictions. Both the model we use for morphological tagging and the lemmat-

[1] In this paper, we use lect as a neutral term for any given language variety, whether it is a standard, a dialect, or a sociolect.

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 167–175
September 11–12, 2023. ©2023 Association for Computational Linguistics

iser (at least in prediction mode) should be able to run on an individual device with no more than 6GB V-RAM (the specs of NVIDIA GeForce GTX1060, currently the most widespread GPU). The training data may vary in size, however, the datasets that we select on distributional properties basis should not exceed 500 000 tokens.

Section 2 contains previous research on the topics of lemmatisation in general and Russian lemmatisation in particular. In section 3, we describe the data. Section 4 includes a description of the method. Section 5 describes the experiments and the analysis of their results. In section 6, we wrap up the research, stating either confirmation or refutation for each of the hypotheses, as well as the possible future directions of the research.

2 Related Work

Currently, there are two predominant approaches to lemmatisation. The first is the classification approach: the model determines the rule of lemmatisation for a given token and then applies the rule (Mills, 1998; Chrupała, 2006; Plisson et al., 2008; Gesmundo and Samardžić, 2012; Radziszewski, 2013). This approach tends to be monolingual (Anastasyev, 2020; Torre Alonso, 2022). The second is the generally multilingual sequence-to-sequence approach when the input (token and its features) is transformed directly to the output sequence, lemma (Straka and Straková, 2017; Bergmanis and Goldwater, 2018; Kanerva et al., 2021).

Russian lemmatisation currently dominates the East Slavic lemmatisation landscape (Anastasyev, 2020), including historical varieties, with both rule-based and automatic methods (Berdičevskis et al., 2016; Pedrazzini and Eckhoff, 2021). However, territorial lects have not yet gained the same kind of attention, while the lemmatisers designed for specific corpora are not open-source (Kryuchkova and Goldin, 2011, 2015). Russian web as corpora and social media texts are included in the evaluation pipelines but generally are not the centre of attention (Sorokin et al., 2017).

There are different ways to enhance the performance of a lemmatisation model, morphological tagging being the most common (Anastasyev, 2020). The ensemble models that enhance lemmatisation efficiency with external resources (Milintsevich and Sirts, 2021) are gaining popularity, especially for historical low-resource territorial lects (de Graaf et al., 2022). And given that social media texts are similar to them (Piotrowski, 2012), the contemporary vocabulary dictionaries are going to be of use in further research.

3 Data

We employ two groups of datasets: the training datasets and the evaluation datasets. Training datasets are generally well-established through Russian NLP and mostly contain standard Russian texts. The evaluation datasets group contains both the well-established ones and the ones that are not yet heavily adopted in the Russian NLP.

The largest training dataset is a collection of different Russian National Corpus[2] texts that vary diachronically (from the 1700s to 2010s), orthographically (containing texts in modern orthography, as well as premodern, used mostly before 1917), and genre-wise (including news, poetry, fiction, and social media). We later refer to this dataset as RNC-sampled. This dataset contains nearly 2 million tokens. It is also the dataset the lemmatisation model trained on. We also employ two subsets of RNC-sampled. The first one is Taiga (Shavrina and Shapovalova, 2017), which aims to represent texts from social media that demonstrate a higher level of variation and colloquiality. Taiga contains 197 000 tokens. The second is SynTagRus (Droganova et al., 2018), the biggest Universal Dependencies Russian corpus, containing fiction, non-fiction and news texts. The original SynTagRus contains 1.5 million tokens, we downsampled it to 195 000 tokens for effective comparison with Taiga.

We use three sets of data for evaluation. The first is the tagged part of the Russian General Internet Corpus (Belikov et al., 2018), designed for MorphRuEval-2017 (Sorokin et al., 2017). It is 270264 tokens in size. Later we refer to this dataset as GIKRYA. GIKRYA consists of different texts from the Internet, which possess a high degree of variation and lack orthographical normalisation. The tagging of the GIKRYA part that we employ is human-checked.

The second evaluation dataset is the scraped collection of tweets from 2022 to 2023, selected based on them containing words мокша ‘Moksha’, эрзя ‘Erzya’, and Саратов ‘Saratov’. The tweets contain texts from the regional mass media, as well as everyday communication, concerning current politics, by speakers of different origins and

[2]ruscorpora.ru

backgrounds. We slightly manually normalised the texts, correcting the most obvious errors, such as *проектрование > проектирование 'design'. The variation degree in this dataset, despite the minor edits, remains high, mostly due to the non-standard compounds, such as иворовал (<ива + воровал 'willow + steal.PAST.3.SG.M'), and non-standard orthography, for instance, расейськая 'Russian'. We provide human-checked lemmata (without PoS/morphological tagging) for this dataset. We later refer to this dataset as MES-Tweets. MES-Tweets contains 6100 tokens.

The third group of evaluation datasets is the transcribed recordings of interviews with speakers of Russian continuum dialects (small territorial lects) Belogornoje (Saratov Region, Russia, southern type, the territory where Russian speakers arrived after the Russian dialect system had formed), and Megra (Vologda Region, Russia, northern type, the territory where Russian speakers had arrived before the split of the Old East Slavic dialect continuum). We take the material for both Belogornoje and Megra (as we refer to them later) from Saratov dialectological corpus (Kryuchkova and Goldin, 2011, 2015). These datasets are in themselves homogeneous, yet they differ from the training datasets, representing small territorial lects, rather than variation within the standard. Belogornoje and Megra together contain 4372 tokens, with Megra being slightly larger (2856 versus 1516 tokens). Both datasets possess gold lemmatisation and morphological tagging, though annotation schema differences make the use of the latter hardly applicable to this study.

We present the short summary for each dataset in Table 1.

4 Method

To determine the degree, to which the morphological properties of a training dataset may influence the lemmatisation efficiency of an evaluation dataset, we present the following experiment pipeline.

Beforehand, we fine-tune the lemmatisation model with the largest morphologically tagged dataset available, RNC-sampled. The lemmatisation model is a sequence-to-sequence one, employing the BART architecture with the largest number of parameters (430M) (Lewis et al., 2020). This model, BART-large, is used for all the lemmatisation experiments.

For part-of-speech tagging, we use the Stanza tagger (Qi et al., 2018, 2020), modified for the low-resource lects (Scherrer, 2021). We train Stanza on three different datasets, RNC-sampled, Taiga, and SynTagRus-downsampled. RNC-sampled has the largest variation degree and the largest size. Taiga, being relatively smaller, consists of social media texts that inherently possess a high degree of variety. SynTagRus-downsampled is comparable in size to Taiga, but it is much more homogeneous genre-wise.

Training yields three taggers for each of the datasets (RNC-sampled, Taiga, and SynTagRus). We then test these models. For this, we use GIKRYA as a dataset both completely independent from the Russian National Corpus and possessing a significant variation degree. This provides us with the preliminary idea of whether the knowledge acquired through RNC-sampled, Taiga, and SynTagRus-downsampled data, may aid the model in tagging a completely different dataset.

Then we perform the three stages of the lemmatisation experiments. As a baseline for each stage, we use two different tactics. The first is a simple token-to-lemma method when each token is taken as its own lemma. The second is using BART-large on bare tokens (with input in the form of [token] [part-of-speech information] [morphological tagging information] and lemma as a desired output). For each stage, we tag the datasets of GIKRYA (stage 1), MES-Tweets (stage 2) and Belogornoje and Megra (stage 3) with each of the morphological tagging models available, providing silver (non-human-checked, yet performed by a model that generally produces satisfactory results) morphological tagging. The stages represent the growing degree of distance between standard Russian and the variations that form the datasets. After that, we lemmatise each of the acquired datasets with BART-large. We compare the results of the lemmatisation against the baseline. As GIKRYA provides the gold morphological tagging, for stage 1 we also lemmatise tokens with gold tagging to set the highest possible bar.

For evaluation, we use accuracy score, combined with different string similarity measures: Levenshtein distance (Levenshtein, 1966), Damerau-Levenshtein distance (Damerau, 1964), and Jaro-Winkler distance (Jaro, 1989; Winkler, 1990). Levenshtein distance that scores additions, deletions, and substitutions of characters gives a more precise picture of sequence-to-sequence model per-

Dataset name	Dataset group	Previous morphological tagging presence	Token number
RNC-sampled	Training	Present	2000000
Taiga	Training	Present	197000
SynTagRus	Training	Present	1500000
GIKRYA	Evaluation	Present	270264
MES-Tweets	Evaluation	Non-present	6100
Belogornoje	Evaluation	Present (different annotation schema)	1516
Megra	Evaluation	Present (different annotation schema)	2856

Table 1: Datasets used in the study

formance in comparison to the accuracy score, reducing the cost of small mistakes and putting the models that generalise over the models that only memorise. Damerau-Levenshtein distance adds substitutions, providing an even more fine-grained picture. Jaro-Winkler distance shows exactly how well models capture the concept of lemmatisation in Slavic languages, favouring the sequences that match from the beginning. We also use normalised versions of these metrics (Grubbs, 1969). Normalisation generally highlights the ability of a model to generalise: if the normalised score is less than its raw counterpart, the model possibly learned to remember particular token-lemma pairs rather than to lemmatise.

5 Experiments and Analysis

We split the experiments into the morphological tagging section and the lemmatisation section. The lemmatisation section consists of three stages. For the first, we use GIKRYA, the web corpus that contains texts of different genres and variations, some further from the standard Russian than others. The second includes the lemmatisation of the MES-Tweets dataset, which possesses a higher variation degree. For the third, we take dialect data, pushing the ability of the models to generalise to the limit.

5.1 Morphological tagging

The morphological tagging results for GIKRYA are in Table 2.

The model trained on RNC-sampled was overfitting. It has the least out-of-vocabulary rate while performing worse than the models trained on Taiga and SynTagRus-downsampled. The model trained on SynTagRus-downsampled performed the best, especially in the exact match category (UFeats). Probably, the homogeneity and the small size of SynTagRus-downsampled allow the model to concentrate on the morphological tagging concept rather than attempting to grasp variation within it. However, all the models achieved relatively high scores, which may make their tagging relevant for the lemmatisation.

5.2 Lemmatisation (GIKRYA)

The results of measuring the efficiency of GIKRYA, morphologically tagged with these models' lemmatisation (later referred to by the name of the dataset we trained them on), are in Table 3.

The results show that gold tagging predictably is the most desired option for the lemmatiser. Models, however, are still able to easily outperform both baselines. The synTagRus-downsampled-trained model demonstrates the highest accuracy score, while the RNC-sampled-trained one shows the highest Jaro-Winkler distance score. Levenshtein and Damerau-Levenshtein distances, including the normalised ones, are the same. Each model helps the lemmatiser to achieve a consistently high score and to understand that Russian lemmata generally start with the same characters as tokens. Importantly, mistakes that the lemmatiser makes are often caused by differences in the lemmatisation policy and not incorrect morphological tagging, cf. регулировать 'control' instead of регулирующий 'the controlling one': in RNC-sampled, Taiga and SynTagRus-downsampled the participles are treated as verbs and lemmatised to an infinitive, while in GIKRYA the participle is a full-fledged part-of-speech category, and the participles are lemmatised to their nominative singular masculine form.

However, the results do not correlate directly with the morphological tagging results, as the RNC-sampled-trained model performs the worst in morphological tagging, yet here it helps the lemmatiser the most to grasp the concept of lemmatisation, and overall to score pretty well. SynTagRus-downsampled-trained model, the best for morpho-

Training dataset	PoS	PoS+Feats	UFeats	OOV
RNC-sampled	83.17	77.65	54.90	**15.59**
Taiga	**85.57**	80.89	54.75	35.03
SynTagRus-downsampled	**85.57**	**82.51**	**60.69**	34.55

Table 2: The efficiency of GIKRYA dataset morphological tagging with Stanza (Qi et al., 2018, 2020; Scherrer, 2021), evaluated by Micro-F1 score, %.The best results here and after are highlighted in **bold**.

Model	A	L	L(N)	D-L	D-L(N)	J-W	J-W(N)
Token-to-lemma	51.79	0.86	0.84	0.86	0.84	93.99	97.21
Bare token	49.07	0.87	0.85	0.87	0.85	86.34	96.5
RNC-sampled	90.41	0.19	0.19	0.19	0.19	98.94	98.94
Taiga	89.93	0.19	0.19	0.19	0.19	98.92	98.92
SynTagRus-downsampled	90.51	0.19	0.19	0.19	0.19	98.93	98.93
Gold	**94.79**	**0.07**	**0.07**	**0.07**	**0.07**	**99.54**	**99.54**

Table 3: The results of GIKRYA lemmatisation evaluation by accuracy score (A, %), raw (L) and normalised(L(N)) Levenshtein, raw (D-L) and normalised (D-L(N)) Damerau-Levenshtein, raw (J-W) and normalised (J-W(N), %) Jaro-Winkler distances.

logical tagging, enables the lemmatiser to do the best in terms of accuracy, but also the latter gets worse Jaro-Winkler results. Only the Taiga-trained model still lags behind.

Morphological tagging mistakes may play some role in the downfalls of the models, for instance, in cases such as аг instead of ага 'yeah', which the tagger treats as a noun in the genitive singular form, misleading lemmatiser that afterwards applies the wrong tactic.

5.3 Lemmatisation (MES-Tweets)

GIKRYA is still a human-checked, heavily normalised dataset. To get the picture of the model's performance in what is functionally terra incognita, we attempt to lemmatise MES-Tweets. The results are in Table 4.

The results differ from the previous experiments. The tagging by the Taiga-trained model aids lemmatiser the most, even if by a slight margin in each given metric. It seems that here the Taiga lemmatisation approach coincides with the target dataset, as it correctly predicts размышляющий 'the thinking one' as the participle lemma in contrast to the infinitive размышлять 'to think', that, for example, SynTagRus lemmatisation rules propose. It also detects some complex nouns, such as финно-угр 'Finno-Ugric', which, for instance, SynTagRus-downsampled-trained model perceives as an adjective, yielding lemma финно-угрый. Morphological tagging yet again heavily defines the dives in the performance of the lemmatiser, but now the Taiga-trained model is seemingly the best with the given dataset. It may be explained by the closeness of the dataset domains: both Taiga and MES-Tweets are social media texts.

To check this, we turn to the dialect datasets, which are close to social media in terms of variation within themselves and when compared to the standard Russian. Results are presented in tables 5 and 6.

5.4 Lemmatisation (Dialect datasets)

Dialect datasets yet again show different results. In Megra, none of the models beat the token-to-lemma baseline by the normalised Jaro-Winkler distance metric, which signals the morphological tagging issues. Incorrect morphological tag detection leads to incorrect sequence-to-sequence transformation, as the confused model applies different rules. For instance, it may predict брести 'to wander' instead of бремя 'burden'. Despite that, Taiga achieves the best score by every other metric.

In Belogornoje, the lemmatiser benefits the most from RNC-sampled-trained model tagging, with Taiga getting close. There may be different factors at play here: the Belogornoje dataset is only a thousand tokens and is closer to the standard Russian, probably, 20th-century fiction, than Megra.

Morphological tagging still seems unable to solve some critical issues. The question of how to treat compound lemmata in the dataset remains, cf. дак и 'so' that is lemmatised only as и 'and' by the model. <ё> is necessary for dialects, though, in the

Model	A	L	L(N)	D-L	D-L(N)	J-W	J-W(N)
Token-to-lemma	58.42	0.71	0.53	0.71	0.53	96.19	98.12
Bare token	53.48	0.77	0.6	0.77	0.6	88.55	97.52
RNC-sampled	86	0.25	0.25	0.25	0.25	98.49	98.49
Taiga	**86.38**	**0.24**	**0.24**	**0.24**	**0.24**	**98.59**	**98.59**
SynTagRus-downsampled	86.1	0.25	0.25	0.25	0.25	98.47	98.47

Table 4: The results of MES-Tweets lemmatisation evaluation.

Model	A	L	L(N)	D-L	D-L(N)	J-W	J-W(N)
Token-to-lemma	60.54	0.86	0.8	0.86	0.8	90.76	**96.8**
Bare token	58.65	0.9	0.84	0.9	0.83	90.3	95.67
RNC-sampled	82.67	0.37	0.37	0.37	0.37	95.65	95.65
Taiga	**83.89**	**0.35**	**0.35**	**0.35**	**0.35**	**95.66**	95.66
SynTagRus-downsampled	81.97	0.4	0.4	0.4	0.4	95.4	95.4

Table 5: The results of Megra dialect lemmatisation evaluation.

standard Russian dataset, it is normalised to <e>. Rare word changing models for verbs like помирать 'to be dying', the forms of which lemmatiser treats as the forms of помереть 'to die' under the influence of more productive models, present the problem as well.

Significant dialect features, for instance, *jakanje*, if shown in lemma, also lead to errors (cf. выдояти 'to milk' that is lemmatised as standard Russian выдоить). Non-standard forms, such as мни 'I-DAT' (cf. standard мне) confuse both the tagger and the lemmatiser, leading to incorrect tagging and subsequent assignment of the token as its lemma, instead of я. But the most significant issue is still the lemmatisation policy, the differences between understanding what should be a lemma for a token in a dataset.

6 Conclusion

The experiments prove that the silver morphological tagging allows a lemmatiser to perform much more efficiently than without any information on morphological tagging (over 40% improvement). We show that silver morphological tagging aids almost as efficiently as gold morphological tagging, lagging only by 4% for web as corpora datasets, such as GIKRYA. This is achieved with the BART-large model, fine-tuned for the standard language. Both the prediction run of BART-large and any training run of modified Stanza (Scherrer, 2021) did not take more than 4 GB GPU on RTX 3060 (mobile). Thus, even if fine-tuning large lemmatiser models themselves on personal computer hardware is still going to remain a focus of further study, morphological tagging and lemmatisation itself may be performed on the relatively small data. The lemmatiser enhanced with data provided by both the Taiga-trained and the SynTagRus-downsampled-trained taggers often performs better than the lemmatiser enhanced with data provided by the RNC-sampled-trained tagger. Even when the situation is opposite, the distance between the results rarely exceeds five per cent.

The hypotheses that we stated at the beginning of the research were the following:

H1: Morphological tagging efficiency directly influences the lemmatisation accuracy.

H2: If the model trains on the larger dataset, the morphological tagging it performs will present stable satisfactory results.

H3: For non-standard data lemmatisation, the distributional properties of the training dataset are generally more important than the sheer size.

The first hypothesis, as GIKRYA experiments show, holds only partially. SynTagRus-downsampled-trained tagger performs the best in terms of morphological tagging, but RNC-sampled-trained tagger performs the best as an aide for the lemmatiser.

The second hypothesis holds: there are no sudden falls in lemmatisation accuracy when the RNC-sampled-trained tagger provides additional data, even if the results achieved are not the best.

The third hypothesis generally holds. The less the dataset resembles the standard Russian, the more efficient becomes the enhancement with data acquired from the Taiga-trained tagger, and the less

Model	A	L	L(N)	D-L	D-L(N)	J-W	J-W(N)
Token-to-lemma	59.37	0.83	0.78	0.83	0.78	92.34	97.1
Bare token	58.05	0.85	0.8	0.85	0.79	92.22	97.16
RNC-sampled	**84.89**	**0.29**	**0.29**	**0.29**	**0.29**	**97.85**	**97.85**
Taiga	83.71	0.31	0.31	0.31	0.31	97.73	97.73
SynTagRus-downsampled	83.25	0.33	0.33	0.33	0.33	97.61	97.61

Table 6: The results of Belogornoje dialect lemmatisation evaluation.

efficient becomes the enhancement with data acquired from the SynTagRus-downsampled-trained tagger. This is because Taiga, social media texts, is much more heterogeneous than SynTagRus-downsampled. Additional morphological information from RNC-sampled-trained tagger run beats the one that Taiga provides, but only for Belogornoje. It is important to remember that parts of Taiga are included in RNC-sampled, and interaction between these parts and other parts of the RNC-sampled enabled the lemmatiser to process Belogornoje especially well. However, this case is an outlier.

The future direction of the research becomes clear: further search for a dataset that provides the best silver morphological tagging for dialect data as well as attempts at efficiently using small transformers (such as TinyBART (Shleifer and Rush, 2020)) that one can fine-tune with personal computer hardware.

References

Dan Anastasyev. 2020. Exploring pretrained models for joint morpho-syntactic parsing of Russian. In *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2020"*, pages 1–12. Moscow.

Vladimir Belikov, Nikolay Kopylov, Alexander Piperski, Vladimir Selegey, and Serge Sharoff. 2018. Big and diverse is beautiful: A large corpus of Russian to study linguistic variation. In *Web as Corpus Workshop (WAC-8)*, pages 24–29, Lancaster, UK. WAC-8 Organising Committee.

Aleksandrs Berdičevskis, Hanna Eckhoff, and Tatiana Gavrilova. 2016. The beginning of a beautiful friendship: rule-based and statistical analysis of Middle Russian. In *Komp'yuternaya lingvistika i intellektual'nye tekhnologii. Trudy mezhdunarodnoj konferencii «Dialog»*, pages 99–111, Moscow, Russia. RSSU.

Toms Bergmanis and Sharon Goldwater. 2018. Context sensitive neural lemmatization with Lematus. 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 1391–1400, New Orleans, Louisiana. Association for Computational Linguistics.

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. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Grzegorz Chrupała. 2006. Simple data-driven context-sensitive lemmatization. *Proces. del Leng. Natural*, 37:121–137.

Fred J. Damerau. 1964. A technique for computer detection and correction of spelling errors. *Commun. ACM*, 7(3):171–176.

Evelien de Graaf, Silvia Stopponi, Jasper K. Bos, Saskia Peels-Matthey, and Malvina Nissim. 2022. AGILe: The first lemmatizer for Ancient Greek inscriptions. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5334–5344, Marseille, France. European Language Resources Association.

Kira Droganova, Olga Lyashevskaya, and Daniel Zeman. 2018. Data conversion and consistency of monolingual corpora: Russian UD treebanks. In *Proceedings of the 17th International Workshop on Treebanks and Linguistic Theories (TLT 2018)*, pages 52–65, Oslo University, Norway. Linköping University Electronic Press.

Andrea Gesmundo and Tanja Samardžić. 2012. Lemmatisation as a tagging task. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 368–372, Jeju Island, Korea. Association for Computational Linguistics.

Frank E. Grubbs. 1969. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21.

Matthew A. Jaro. 1989. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *Journal of the American Statistical Association*, 84(406):414–420.

Jenna Kanerva, Filip Ginter, and Tapio Salakoski. 2021. Universal Lemmatizer: A sequence-to-sequence model for lemmatizing Universal Dependencies treebanks. *Natural Language Engineering*, 27(5):545–574.

Steven Krauwer. 1998. ELSNET and ELRA: Common past, common future. *ELRA Newsletter*, 3(2).

Steven Krauwer. 2003. The Basic Language Resource Kit (BLARK) as the first milestone for the language resources roadmap. In *Proceedings of the International Workshop "Speech and Computer", SPECOM 2003*, pages 8–15, Moscow, Russia. Moscow State Linguistic University.

Olga Kryuchkova and Valentin Goldin. 2011. Corpus of Russian dialect speech: concept and parameters of evaluation. In *Computational Linguistics and Intellectual Technologies. Proceedings of International Conference "Dialog–2011"*, pages 359–367, Moscow, Russia. RSSU.

Olga Kryuchkova and Valentin Goldin. 2015. The parameters of text processing for the Russian dialect corpus. In *Proceedings of the international conference "Corpus linguistics — 2015"*, pages 307–314, Saint Petersburg, Russia. SPbU.

Vladimir Iosifovich Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

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.

Kirill Milintsevich and Kairit Sirts. 2021. Enhancing sequence-to-sequence neural lemmatization with external resources. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3112–3122, Online. Association for Computational Linguistics.

Jon Mills. 1998. Lemmatisation of the corpus of Cornish. In *Workshop on Language Resources for European Minority Languages, LREC First International Conference on Language Resources and Evaluation*, pages 1–6, Granada, Spain.

Nilo Pedrazzini and Hanne Martine Eckhoff. 2021. OldSlavNet: A scalable Early Slavic dependency parser trained on modern language data. *Software Impacts*, 8:100063.

Michael Piotrowski. 2012. *Natural Language Processing for Historical Texts*. Springer.

Joël Plisson, Nada Lavrac, Dunja Mladenić, and Tomaž Erjavec. 2008. Ripple down rule learning for automated word lemmatisation. *AI Commun.*, 21:15–26.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2018. Universal Dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, 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.

Adam Radziszewski. 2013. Learning to lemmatise Polish noun phrases. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 701–709, Sofia, Bulgaria. Association for Computational Linguistics.

Yves Scherrer. 2021. Adaptation of morphosyntactic taggers. In *Similar Languages, Varieties, and Dialects: A Computational Perspective*, Studies in Natural Language Processing, page 138–166. Cambridge University Press.

Tatiana Shavrina and Olga Shapovalova. 2017. To the methodology of corpus construction for machine learning: «Taiga» syntax tree corpus and parser. In *Proceedings of the International Conference "CORPORA 2017"*, Saint-Petersbourg, Russia.

Sam Shleifer and Alexander M. Rush. 2020. Pre-trained summarization distillation.

Alexey Sorokin, Tatiana Shavrina, Olga Lyashevskaya, Victor Bocharov, Svetlana Alexeeva, Kira Droganova, Alena Fenogenova, and Dmitry Granovsky. 2017. MorphoRuEval-2017: an evaluation track for the automatic morphological analysis methods for Russian. In *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2017"*, pages 1–17, Moscow, Russia. RSSU.

Milan Straka and Jana Straková. 2017. Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *ArXiv*, abs/1409.3215.

Roberto Torre Alonso. 2022. Automatic lemmatization of Old English class III strong verbs (L-Y) with ALOEV3. *Journal of English Studies*, 20:237–266.

William E. Winkler. 1990. String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage. In *Proceedings of the Section on Survey Research*, pages 354–359.

Improving BERT Pretraining with Syntactic Supervision

Giorgos Tziafas
University of Groningen
g.t.tziafas@rug.nl

Konstantinos Kogkalidis
Aalto University
kokos.kogkalidis@aalto.fi

Gijs Wijnholds
Leiden Institute of Advanced Computer Science
g.j.wijnholds@liacs.leidenuniv.nl

Michael Moortgat
Utrecht University
m.j.moortgat@uu.nl

Abstract

Bidirectional masked Transformers have become the core theme in the current NLP landscape. Despite their impressive benchmarks, a recurring theme in recent research has been to question such models' capacity for syntactic generalization. In this work, we seek to address this question by adding a supervised, token-level supertagging objective to standard unsupervised pretraining, enabling the explicit incorporation of syntactic biases into the network's training dynamics. Our approach is straightforward to implement, induces a marginal computational overhead and is general enough to adapt to a variety of settings. We apply our methodology on Lassy Large, an automatically annotated corpus of written Dutch. Our experiments suggest that our syntax-aware model performs on par with established baselines, despite Lassy Large being one order of magnitude smaller than commonly used corpora.

1 Introduction

In recent years, the advent of Transformers (Vaswani et al., 2017) has paved the way for high-performing neural language models, with BERT (Devlin et al., 2019) and its many variants being the main exemplar (Liu et al., 2019; Sanh et al., 2019; Lan et al., 2020). BERT-like models achieve state-of-the-art scores in most major NLP benchmarks via a two-step process. First, they are trained on massive-scale, minimally processed raw text corpora by employing the so-called masked language modeling (*MLM*) objective. Task-specific refinements are then obtained by fine-tuning the pretrained model on labeled corpora, usually orders of magnitude smaller in size.

This pipeline, despite its attested performance, suffers from two key limitations. On the one hand, training a BERT-like model from scratch requires an often prohibitive amount of data and computational resources, barring entry to research projects that lack access to either. On the other hand, a naturally emerging question is whether such models develop an internal notion of syntax. Discovery of structural biases is hindered by their distributed, opaque representations, requiring manually designed *probing* tasks to extract evidence of syntactic awareness (Hewitt and Manning, 2019; Tenney et al., 2019; Kim et al., 2020; Clark et al., 2019a; Goldberg, 2019; Hu et al., 2020). Alternatively, when syntactic evaluation becomes the focal point, it is usually deferred to downstream tasks (Kitaev et al., 2019; Zhang et al., 2020a), owing both to the lack of sufficiently large labeled corpora as well as the computational bottleneck imposed by hard-to-parallelize operations.

In this work, we seek to alleviate both points by considering them in tandem. Contrary to prior work, we consider the case of introducing explicit syntactic supervision during the pretraining process and investigate whether it can allow for a reduction in the data needs of a BERT-like language model. To facilitate this, we couple the standard unsupervised MLM task with a supervised task, mapping each distinct word to a *supertag*, an abstract syntactic descriptor of its functional role within the context of its surrounding phrase. In essence, this amounts to simple token-level classification, akin to traditional supertagging (Bangalore and Joshi, 1999), except for parts of the input now being masked. In employing both objectives, we ensure that our model is syntax-aware by construction, while incurring only a negligible computational overhead. We evaluate the trained model's performance in a variety of downstream tasks and find that it performs on par with established models, despite being trained on a significantly smaller corpus. Our preliminary experiments suggest an improvement to pretraining robustness and offer a promising direction for cheaper and faster training of structure-enhanced language models. Reflecting on the added objective, we call our model *tagBERT*.

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 176–184
September 11–12, 2023.

2 Background

Embedding structural biases in neural language models has been a key theme in recent research. Most syntax-oriented models rely on computationally intensive, hard-to-parallelize operations that constrain their integrability with the state of the art in unsupervised language modeling (Tai et al., 2015; Dyer et al., 2016; Kim et al., 2019). This can be ameliorated by either asynchronous pretraining, relying on accurate but slow oracles (Kuncoro et al., 2019), or multi-task training, where the system is exposed to a syntactic task for only part of its training routine (Clark et al., 2018, 2019b). In the BERT setting, there have been attempts at modifying the architecture by either overlaying syntactic structure directly on the attention layers of the network (Wang et al., 2019b) or imposing shallow syntactic cues and/or semantic information in a multi-task setting (Zhang et al., 2020b; Zhou et al., 2020). While such a setup allows for efficient parallel pretraining, the rudimentary nature of the utilized annotations typically forfeits fine aspects of sentential structure, such as function-argument relations.

In this paper, we adopt lexicalism in the categorial grammar tradition (Ajdukiewicz, 1935; Lambek, 1958; Buszkowski et al., 1988; Steedman, 1993; Moortgat, 1997), according to which (most of) the grammatical structure of a language is encoded in its lexicon via an algebra of types that governs the process of phrasal composition. Under such a regime, the parse tree underlying a sentence can be partially (or even fully, in the case of an adequately "strict" grammar) recovered from its constituent words and their respective types alone. In applied terms, the lexical nature of categorial grammars provides us with the opportunity of capturing syntax in a fully-parallel fashion that is straightforward to incorporate with the masked language modeling objective of BERT-like architectures, a fact so far generally overlooked by machine learning practitioners. This perspective is in line with recent insights arguing for the necessity of explicit supervision for syntactic acquisition (Bailly and Gábor, 2020).

The only prerequisite for our methodology is an adequately sized, categorially annotated corpus. Even though gold standard corpora exist for a variety of languages and grammars (Chen and Shanker, 2004; Hockenmaier, 2006; Hockenmaier and Steedman, 2007; Tse and Curran, 2010; Ambati et al., 2018; Kogkalidis et al., 2020b), their size is generally insufficient for training a parameter-rich neural language model. This limiting factor can be counteracted by either lexicalizing existing silver-standard corpora of a larger size, or by using an off-the-shelf, high-performance supertagger to annotate the source data prior to pretraining. In both cases the trained system is likely to inherit common errors of the data-generating teacher; the question is whether the added structural biases facilitate faster training of more general language models, despite potential tagging inaccuracies.

3 Methodology

3.1 Data

To facilitate both the data needs of the neural language model and the added supertagging objective we employ Lassy Large (van Noord et al., 2013), a corpus of written Dutch, automatically parsed using the Alpino parser (Bouma et al., 2001). The dataset is comprised of a selection of smaller corpora from varying sources, ranging from excerpts from conventional and modern media to spoken transcripts, enumerating a total of almost 800 million words. Lassy's syntactic analyses take the form of directed acyclic graphs, with nodes corresponding to words or phrases marked with their part-of-speech as well as syntactic category labels and edges denoting dependency relations. To make the analyses applicable for our setup, we lexicalize them using the type extraction algorithm of Kogkalidis et al. (2020b). The algorithm traverses a parse graph and encodes its structure in a linear logic proof, under the general paradigm of categorial type logics (Moortgat, 1997), simultaneously capturing function-argument and dependency structure. Words, i.e. fringe nodes in the graph, are assigned *types*, abstract syntactic signs that encode a considerable portion of the full structure.

Applying the extraction algorithm, we obtain a collection of around 66 million sentences, represented as sequences of word-type pairs. We drop about 20 million of these in a sanitation step, due to either being duplicates or overlapping with any of the evaluation tasks. We tokenize words using a preconstructed WordPiece (Schuster and Nakajima, 2012) vocabulary of 30 000 tokens based on a larger collection of written Dutch corpora (Vries et al., 2019). Further, we keep the 2 883 most frequent types, which suffice to cover 95% of the type occurrences in the dataset, and replace the filtered out types with an UNK token. We finally discard sentences lying in the 5%-tail of the length distribution, and train with 45 million sentences spanning less than 100 sub-word tokens.

3.2 Model

Our model is a faithful replica of $\text{BERT}_{\text{BASE}}$, except for having a hidden size of 1 536 instead of 3 072 for the intermediate fully-connected layers, reducing our total parameters from 110 to 79 million. We further employ a linear projection from the model's dimensionality to the number of types in our vocabulary, which we attach to the output of a prespecified encoder block. The projection can be separably applied on the encoder's intermediate representations, allowing us to optionally query the model for a class weighting over types for each input token.

This addition accounts to a mere 2.5% of the model's total parameter count and only incurs a negligible computational overhead if explicitly enabled, as it does not interfere with the forward pass when the system is run solely as a contextualization model. If the type classification layer is enabled during pretraining, it introduces a clear error signal that updates all network weights up to the connected encoder block, bolstering the correct acquisition of syntax in the bottom part of the encoding pipeline.

3.3 Pretraining

To train our model, we feed it partially masked sentences following the methodology of Liu et al. (2019); we dynamically mask continuous spans of tokens belonging to the same word and drop the next sentence prediction task, training on single sentences instead. Attaching the type classification layer at the fourth encoder block, we end up with two output streams.[1] One is a prediction over the subword vocabulary for each masked token, as in vanilla BERT, whereas the other comes from the type classifier, yielding a prediction over the type vocabulary for every token, masked or otherwise.[2] We obtain a loss function by summing the cross-entropy between predictions and truths for each output stream.

To deal with the misalignment between subword units and types, we associate every type with the first token of its corresponding word, and mask out predictions spanning subsequent tokens when performing the loss computation. Similarly, we do not penalize predictions over types discarded by the occurrence count filtering (UNK types). For regularization purposes, we randomly replace output types 1% of the time (Wu et al., 2019).

Following standard practices, we optimize using AdamW (Loshchilov and Hutter, 2019) with a batch size of 256, shuffling and iterating the dataset 8 times. The learning rate is gradually increased to 10^{-4} over 10 000 steps and then decayed to zero using a linear warm-up and decay schedule.

4 Evaluation

To evaluate the trained model, we measure its performance on the below selection of downstream tasks, after fine-tuning. We keep our fine-tuning setup as barebones as possible, using Adam (Kingma and Ba, 2014) with a batch size of 32 and a learning rate of 3×10^{-5}. We apply model selection based on the validation-set performance and report test-set results (averaged over three runs) against the available baselines of each task in Table 1. In order to provide fair comparisons, we replicate the evaluation of other models using the same experimental setup.

Lassy Small is a gold-standard syntactically annotated corpus for written Dutch (van Noord et al., 2013). We fine-tune a POS tagger on the subset of the corpus that has been converted to Universal Dependency format (Bouma and van Noord, 2017).

SoNaR-1 is a curated subset of Lassy Small that includes several layers of manually added annotations (Delaere et al., 2009). We employ the named entity recognition (NER), part-of-speech (POS), semantic role labeling (SRL) and spatio-temporal relation tags (STR) that come packed with the corpus and treat their classification as downstream tasks. NER contains approximately 60 000 samples and 6 class labels encoded in the IOB scheme. POS tagging contains about 16 000 samples and comes in two varieties: coarse (12 classes) and fine-grained (241 classes, out of which only 223 appear in the training data, many just once). SRL comes also in two varieties: a) predicate-argument structures, encoded with IOB scheme, and b) modifiers, enhanced with modified phrase labels, for a total of

[1]The choice of depth for the type classifier is due to preliminary experiments where we let a trainable layer weighter freely select from the range of encoder blocks. In the vast majority of runs, most of the importance was interestingly assigned to the fourth layer.

[2]Masking entire words for the supertagging task can be seen as a severe form of regularization, à la channel dropout.

	SoNaR-1			*Lassy UD*	*CoNLL*	*Æthel*	
	POS-coarse	POS-fine	NER	POS	NER	Supertags	Parse
BERTje (Vries et al., 2019)	**98.8**	**97.5**	**87.4**	96.4	**90.6**	85.5	56.9
RobBERT (Delobelle et al., 2020)	98.5	97.2	84.8	96.2	85.9	86.3	56.8
tagBERT (ours)	**98.8**	97.4	87.0	**96.7**	89.9	**86.6**	**58.3**

Table 1: Comparative performance for a selection of downstream tasks. We report test set accuracy (%) on all tasks except NER, where we report F1 scores (%) as produced by the CoNLL evaluation script (Tjong Kim Sang, 2002). For a fair comparison, we replicate the fine-tuning process on all pretrained baselines, including truncation of the maximum token length to 100.

	SoNaR-1			*Europarl*	*DBRD*
	SRL-pred	SRL-mod	STR	die/dat	sentiment
BERTje (Vries et al., 2019)	85.3	67.2	57.3	95.0	93.0
RobBERT (Delobelle et al., 2020)	–	–	–	98.7	**95.1**
tagBERT (ours)	**86.5**	**67.8**	**68.0**	**99.1**	93.8

Table 2: Comparative performance on higher-level downstreak tasks. Scores are F1 (%) for SRL/STR, and test set accuracy (%) for *die/dat* disambiguation and sentiment analysis. –: no results available.

30 000 samples. STR contains a total of 58 000 spatio-temporal tags, including geolocations and use of past verbe tense.

CoNLL-2002 is a named entity recognition dataset from the corresponding shared task (Tjong Kim Sang, 2002). The dataset contains 4 class labels, also encoded in the IOB scheme, with a total size of approximately 24 000 samples.

Æthel is a typelogical derivation dataset, generated by applying the type extraction algorithm to Lassy Small (Kogkalidis et al., 2020b). We replicate the experiments of Kogkalidis et al. (2020a) to train a typelogical grammar parser, but instantiate the encoder part with the baselines of Table 1, and report token-level supertagging accuracy as well as full sentential parsing accuracy in the greedy setting. We note that even though our model is exposed to types during pretraining, their representation format is vastly different during the fine-tuning process; rather than being classification outputs for each word, they are broken down to their primitive symbols and transduced from the input sequence with auto-regressive *seq2seq* decoding. In that sense, this task helps us assess the generality of the learned representations.

Dutch Europarl is a sanitized subset of transcripts of the European Parliament in Dutch (Koehn, 2005), used for zero-shot evaluation of the task of relative pronoun disambiguation. The task revolves around picking the most likely between the Dutch relative pronouns *die* and *dat*. While the two agree in their syntactic function (and grammatical category), the former selects exclusively for gendered nouns, whereas the latter selects for neuter ones. As such, the task measures our model's capacity to resolve morphosyntactic constraints in the presence of grammatical category invariants. This corpus enumerates a total of 1.56M sentences, 90.7% of which contain at least one relative pronoun.

110k Dutch Book Reviews Dataset. The Dutch Book Reviews Dataset (DBRD) is a sentiment analysis bechnmark which comprises around 110k Dutch book reviews taken from hebban.nl, out of which 22 252 are manually labeled as either positive or negative (Van der Burgh and Verberne, 2019) and segmented into 90% training and 10% testing splits. Unlike previous tasks, sentiment analysis is done on the sentential, rather than token level, serving as a measure of the model's semantic understanding.

5 Discussion

Our model performs on par across all tasks considered, indicating pretraining robustness comparable to the heavy weight baselines of BERT- (Devlin et al., 2019) and RoBERTa-based (Liu et al., 2019) models.[3] Considering the non-ideal nature of the silver-standard tags, the significantly smaller size of our corpus compared to competing models, and the ca. 30% reduced parameter count (79M against 119M for BERTje and 117M for RobBERT), our results can be seen as strong evidence in favor of explicitly encoding structural biases in the pretraining process of neural language models. Opting for a lexicalized representation of structure allows for a seamless and cost-efficient integration with BERT's core architecture, essentially removing the computational bottleneck of alternating between tensor optimization and structure manipulation.

6 Conclusion

We introduced tagBERT, a variation of BERT that is biased towards syntax through coupling the standard MLM loss with a supertagging objective. We trained tagBERT on a modestly sized, silver-standard corpus of written Dutch – after first lexicalizing its annotations – and evaluated the trained model on a number of downstream NLP tasks after fine-tuning. Despite a reduced parameter count and the corpus' modest size, our method is achieving performance comparable to established state-of-the art models. This result is contrary to the ongoing trend of utilizing increasingly more data and augmenting model capacity, instead suggesting potential benefits from incorporating richer annotations in convenient representation formats. Our work aims towards a syntactically-transparent, cost-efficient language model that combines both the rigor of formal linguistic theories and the representational power of large-scale unsupervised learning.

Retroactive Placement The current work is situated in the historical landscape where probing for syntactic awareness and the possibility of injecting syntactic structures in the network were still novel enterprises. Alongside and following our original endeavour, many more studies have investigated the role of syntax and ways to incorporate it within large language models, with the end-goal of either jointly acquiring the two, or of using explicit syntactic guidelines to constrain language generation (Zanzotto et al., 2020; Sartran et al., 2022; Li et al., 2021; Bai et al., 2021; Song et al., 2022; Xie et al., 2021; Li et al., 2023). Other than technical differences on the neural front, our work diverges in opting for a linearized representation of syntax through categorial grammars. This choice stands out for its elegance and formal coherence, setting it apart from more widely used alternatives like constituency trees and dependency arcs. Indeed, categorial grammars (regardless of the particular flavor adopted) offer the means for an expressive, yet fully lexicalized, modeling of syntax and compositional meaning. Their integration into the modern NLP toolkit is facilitated by this inherent flexibility, offering the potential for intricate interplay between structure and form – a potential that still remains, for the most part, untapped.

Future work Given the embedding of this paper within the landscape concerning syntactic awareness and large language models, future work would be based on recent developments in neurosymbolic approaches to lexicalized grammar formalisms. For instance, recent developments in neural supertagging could be exploited, for instance by updating the supertagging from the discriminative setting to a constructive one (Prange et al., 2021; Kogkalidis et al., 2023).

Besides developments relevant to model architecture, several novel evaluation tasks for Dutch have been developed that may shed light on the distinction between vanilla Transformer-based models and syntactically informed ones. For example, the two-sentence classification task of Natural Language Inference (NLI) is a typical task that tests for lexical, syntactic, and sentence-level understanding, for which two Dutch benchmarks exist (Wijnholds and Moortgat, 2021; Wijnholds, 2023). A comparison between a vanilla model, the syntactically informed tagBERT, and the neurosymbolic approach of Abzianidze and Kogkalidis (2021) is in place to put the relationship of syntax and NLI in perspective. Further on the Dutch front, we would be keen to test the model's ability to understand discontinuous verb-subject dependencies as in Kogkalidis and Wijnholds (2022), or to disambiguate relative clauses as in Wijnholds and Moortgat (2023).

Finally, we invite and look forward to different research directions, such as experimentation with different languages and grammar formalisms, integration with existing pre-trained models in an intermediate-training fashion (Wang et al., 2019a) and exploring architectural adjustments that would allow a two-way dependence or a stronger interfacing between the lexical and syntactic modalities.

[3] Implementation code is available at `https://github.com/gtziafas/type-enhanced-language-modeling`.

7 Acknowledgments

The authors warmly express their support from the Dutch Research Council (NWO) due to which three of the authors were operational under the scope of the project “A composition calculus for vector-based semantic modelling with a localization for Dutch" (360-89-070), at the time of execution of this research project.

References

Lasha Abzianidze and Konstantinos Kogkalidis. 2021. A logic-based framework for natural language inference in dutch. *Computational Linguistics in the Netherlands Journal*, 11:35–58.

Kazimierz Ajdukiewicz. 1935. Die syntaktische Konnexität. *Studia philosophica*, 1:1–27. English translation “Syntactic Connexion" by H. Weber in McCall, S. (Ed.) *Polish Logic*, pp. 207–231, Oxford University Press, Oxford, 1967.

Bharat Ram Ambati, Tejaswini Deoskar, and Mark Steedman. 2018. Hindi CCGbank: A CCG treebank from the Hindi dependency treebank. *Language Resources and Evaluation*, 52(1):67–100.

Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, Jing Yu, and Yunhai Tong. 2021. Syntax-BERT: Improving pre-trained transformers with syntax trees. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3011–3020, Online. Association for Computational Linguistics.

Raphaël Bailly and Kata Gábor. 2020. Emergence of syntax needs minimal supervision. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 477–487, Online. Association for Computational Linguistics.

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):237–265.

Gosse Bouma and Gertjan van Noord. 2017. Increasing return on annotation investment: The automatic construction of a universal dependency treebank for Dutch. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 19–26, Gothenburg, Sweden. Association for Computational Linguistics.

Gosse Bouma, Gertjan Van Noord, and Robert Malouf. 2001. Alpino: Wide-coverage computational analysis of Dutch. In *Computational linguistics in the Netherlands 2000*, pages 45–59. Brill Rodopi.

Wojciech Buszkowski, Witold Marciszewski, and Johan van Benthem. 1988. *Categorial grammar*. John Benjamins Publishing.

John Chen and Vijay K Shanker. 2004. Automated extraction of TAGs from the Penn treebank. In *New developments in parsing technology*, pages 73–89. Springer.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019a. What does BERT look at? An analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. 2019b. BAM! born-again multi-task networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5931–5937, Florence, Italy. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925, Brussels, Belgium. Association for Computational Linguistics.

Isabelle Delaere, Veronique Hoste, and Paola Monachesi. 2009. Cultivating trees: Adding several semantic layers to the Lassy treebank in SoNaR. In *7th International workshop on Treebanks and Linguistic Theories (TLT-7)*, pages 135–146. LOT (Landelijke Onderzoekschool Taalwetenschap).

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: a Dutch RoBERTa-based language model. *arXiv preprint arXiv:2001.06286*.

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.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California. Association for Computational Linguistics.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Julia Hockenmaier. 2006. Creating a CCGbank and a wide-coverage CCG lexicon for German. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 505–512, Sydney, Australia. Association for Computational Linguistics.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Taeuk Kim, Jihun Choi, Daniel Edmiston, and Sang-goo Lee. 2020. Are pre-trained language models aware of phrases? Simple but strong baselines for grammar induction. In *International Conference on Learning Representations*.

Yoon Kim, Alexander Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. 2019. Unsupervised recurrent neural network grammars. 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 1105–1117, Minneapolis, Minnesota. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. 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.

Konstantinos Kogkalidis, Michael Moortgat, and Richard Moot. 2020a. Neural proof nets. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 26–40, Online. Association for Computational Linguistics.

Konstantinos Kogkalidis, Michael Moortgat, and Richard Moot. 2020b. Æthel: Automatically extracted typelogical derivations for Dutch. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5257–5266, Marseille, France. European Language Resources Association.

Konstantinos Kogkalidis, Michael Moortgat, and Richard Moot. 2023. SPINDLE: Spinning raw text into lambda terms with graph attention. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 128–135, Dubrovnik, Croatia. Association for Computational Linguistics.

Konstantinos Kogkalidis and Gijs Wijnholds. 2022. Discontinuous constituency and BERT: A case study of Dutch. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3776–3785, Dublin, Ireland. Association for Computational Linguistics.

Adhiguna Kuncoro, Chris Dyer, Laura Rimell, Stephen Clark, and Phil Blunsom. 2019. Scalable syntax-aware language models using knowledge distillation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3472–3484, Florence, Italy. Association for Computational Linguistics.

Joachim Lambek. 1958. The mathematics of sentence structure. *The American Mathematical Monthly*, 65(3):154–170.

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*.

Yafu Li, Leyang Cui, Jianhao Yan, Yongjing Yin, Wei Bi, Shuming Shi, and Yue Zhang. 2023. Explicit syntactic guidance for neural text generation.

Zhongli Li, Qingyu Zhou, Chao Li, Ke Xu, and Yunbo Cao. 2021. Improving BERT with syntax-aware local attention. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 645–653, 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 preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Michael Moortgat. 1997. Categorial type logics. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of logic and language*, chapter 2, pages 93–177. Elsevier/MIT Press.

Jakob Prange, Nathan Schneider, and Vivek Srikumar. 2021. Supertagging the Long Tail with Tree-Structured Decoding of Complex Categories. *Transactions of the Association for Computational Linguistics*, 9:243–260.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Laurent Sartran, Samuel Barrett, Adhiguna Kuncoro, Miloš Stanojević, Phil Blunsom, and Chris Dyer. 2022. Transformer Grammars: Augmenting Transformer Language Models with Syntactic Inductive Biases at Scale. *Transactions of the Association for Computational Linguistics*, 10:1423–1439.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE.

Jian Song, Di Liang, Rumei Li, Yuntao Li, Sirui Wang, Minlong Peng, Wei Wu, and Yongxin Yu. 2022. Improving semantic matching through dependency-enhanced pre-trained model with adaptive fusion. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 45–57, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mark Steedman. 1993. Categorial grammar. *Lingua*, 90(3):221–258.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. 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 1556–1566, Beijing, China. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Daniel Tse and James R. Curran. 2010. Chinese CCGbank: extracting CCG derivations from the Penn Chinese treebank. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1083–1091, Beijing, China. Coling 2010 Organizing Committee.

Benjamin Van der Burgh and Suzan Verberne. 2019. The merits of universal language model fine-tuning for small datasets–a case with dutch book reviews. *arXiv preprint arXiv:1910.00896*.

Gertjan van Noord, Gosse Bouma, Frank Van Eynde, Daniël de Kok, Jelmer van der Linde, Ineke Schuurman, Erik Tjong Kim Sang, and Vincent Vandeghinste. 2013. *Large Scale Syntactic Annotation of Written Dutch: Lassy*, pages 147–164. Springer Berlin Heidelberg.

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, pages 5998–6008. Curran Associates, Inc.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT model.

Alex Wang, Jan Hula, Patrick Xia, Raghavendra Pappagari, R. Thomas McCoy, Roma Patel, Najoung Kim, Ian Tenney, Yinghui Huang, Katherin Yu, Shuning Jin, Berlin Chen, Benjamin Van Durme, Edouard Grave, Ellie Pavlick, and Samuel R. Bowman. 2019a. Can you tell me how to get past sesame street? sentence-level pretraining beyond language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4465–4476, Florence, Italy. Association for Computational Linguistics.

Yaushian Wang, Hung-Yi Lee, and Yun-Nung Chen. 2019b. Tree transformer: Integrating tree structures into self-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 1061–1070, Hong Kong, China. Association for Computational Linguistics.

Gijs Wijnholds. 2023. Assessing monotonicity reasoning in Dutch through natural language inference. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1494–1500, Dubrovnik, Croatia. Association for Computational Linguistics.

Gijs Wijnholds and Michael Moortgat. 2021. SICK-NL: A dataset for Dutch natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1474–1479, Online. Association for Computational Linguistics.

Gijs Wijnholds and Michael Moortgat. 2023. Structural ambiguity and its disambiguation in language model based parsers: the case of dutch clause relativization.

Liwei Wu, Shuqing Li, Cho-Jui Hsieh, and James L Sharpnack. 2019. Stochastic shared embeddings: Data-driven regularization of embedding layers. In *Advances in Neural Information Processing Systems*, volume 32, pages 24–34. Curran Associates, Inc.

Yikuan Xie, Wenyong Wang, Mingqian Du, and Qing He. 2021. Transformer with syntactic position encoding for machine translation. In *Proceedings of* *the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1536–1544, Held Online. INCOMA Ltd.

Fabio Massimo Zanzotto, Andrea Santilli, Leonardo Ranaldi, Dario Onorati, Pierfrancesco Tommasino, and Francesca Fallucchi. 2020. KERMIT: Complementing transformer architectures with encoders of explicit syntactic interpretations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 256–267, Online. Association for Computational Linguistics.

Yu Zhang, Houquan Zhou, and Zhenghua Li. 2020a. Fast and accurate neural CRF constituency parsing. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4046–4053. International Joint Conferences on Artificial Intelligence Organization. Main track.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020b. Semantics-aware BERT for language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9628–9635.

Junru Zhou, Zhuosheng Zhang, Hai Zhao, and Shuailiang Zhang. 2020. LIMIT-BERT: Linguistics informed multi-task BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4450–4461, Online. Association for Computational Linguistics.

MAP: Low-data Regime Multimodal Learning with Adapter-based Pre-training and Prompting

Wenyan Li *
University of Copenhagen
weli@di.ku.dk

Dong Li *
Shanghai AI Lab
lidong@ailab.org.cn

Wanjing Li *
Northeastern University (China)
Leawnn@163.com

Yuanjie Wang *
Beijing Institute of Technology
njwangyuanjie@163.com

Hai Jie *
Beihang University
jsea@buaa.edu.cn

Yiran Zhong *
Shanghai AI Lab
zhongyiran@ailab.org.cn

Abstract

Pretrained vision-language (VL) models have shown impressive results on various multi-modal downstream tasks recently. Many of the benchmark models build on pretrained causal language models (LMs), leveraging the original few-shot learning and generalization capability of the LMs trained with large text corpora. However, these models are often gigantic and require large-scale image and text data with high computational cost to train. This paper introduces a moderate-size model called MAP for efficient VL transfer learning through adapter-based pretraining and prompting. We aim to answer the question of how much we can complete through VL pretraining within the low-data regime while maximizing efficiency in transferring knowledge of a moderate-size frozen LM. Our experiments demonstrate that MAP achieves substantially better zero-shot and few-shot performance on downstream VL tasks with only 10% the size of pretraining data and a 30× lighter pretrained LM backbone compared to Frozen. MAP also outperforms fully trained models of comparable size at retaining its transfer learning ability when the amount of training data reduces.

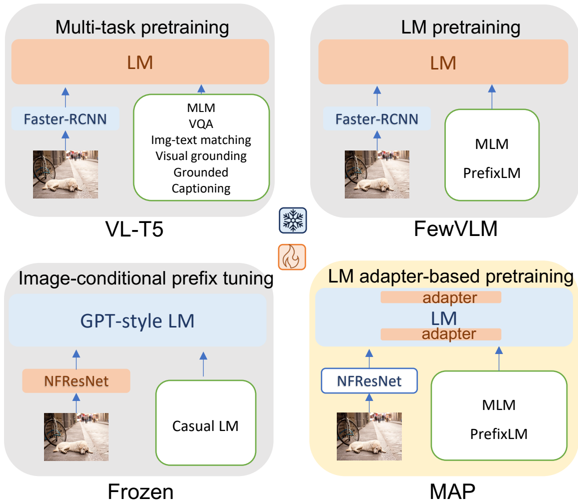


Figure 1: Comparison between VL learning with multitask pretraining, LM pretraining, image-conditional prefix tuning and adapter-based LM pretraining.

1 Introduction

Recent vision-language models commonly leverage pre-trained language models (LMs) on various multimodal tasks. It is crucial for them to retain the original generation capability of the LMs while efficiently incorporating the knowledge from new modalities. A line of work has shown impressive generalization and transfer ability of vision-language models that build on large causal decoder-only LMs (Alayrac et al., 2022; Tsimpoukelli et al., 2021; Eichenberg et al., 2022; Wang et al., 2021a). While powerful, these GPT-style pretrained LMs request high computing machines for deployment.

Compared to decoder-only LMs, Wang et al. (2021c) shows that encoder-decoder model introduces an inductive bias that decouples multimodal feature encoding from generation, yielding improved performance on downstream tasks. Recent research by Jin et al. (2021) also demonstrates that VL models pretrained with a moderate-size encoder-decoder LM backbone can be strong few-shot learners. However, in these approaches, the parameters of the language model were entirely updated while learning vision inputs (Cho et al., 2021; Jin et al., 2021; Wang et al., 2021c), or involving more task-specific data during multitask pretraining or fine-tuning (Sung et al., 2022; Cho et al., 2021).

Naturally, taking account of both model structure and pretraining efficiency, we improve on previous models and introduce an encoder-decoder parameter-efficient VL model, MAP. As shown in Figure 1, we apply adapters for VL pretraining with masked language modeling (Masked LM) and prefix language modeling (PrefixLM) objectives. We keep the backbone encoder-decoder language

*Work done while employed at SenseTime

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 185–190
September 11–12, 2023.

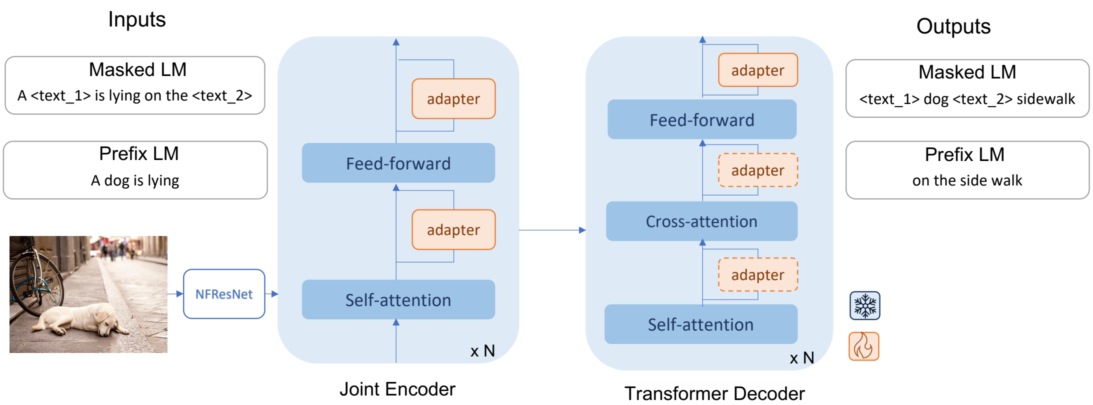


Figure 2: Illustration of the VL pretraining process and the model structure of MAP. We experiment with settings on pretraining by only updating adapters or updating both adapters and the NFResNet image encoder. In the decoder, we experiment with settings of adding adapters to self-attention and cross-attention.

model frozen. In downstream tasks, we provide task-specific prompts to guide the pretrained model for few-shot learning. Our moderate-size model substantially outperforms Frozen on both zero-shot and few-shot learning while using only 10% of multimodal data and a frozen T5-base LM backbone for pretraining.

2 Related Work

Recent work has shown impressive generalization and transfer ability of vision-language models that build on huge pretrained auto-regressive LMs (Alayrac et al., 2022; Tsimpoukelli et al., 2021; Eichenberg et al., 2022; Baevski et al., 2022). Frozen (Tsimpoukelli et al., 2021) updated an NFResNet encoder to create visual prefixes for the frozen LM, transferring the few-shot learning ability of the LM to a multimodal setting. MAGMA (Eichenberg et al., 2022) improved on the results of Frozen by incorporating adapter-based pretraining and a 25M image-text dataset including downstream data into pretraining. Luo et al. (2022) used cross-modal attention for encoding visual and text inputs. The Flamingo model (Alayrac et al., 2022) reached SOTA performance on few-shot VL tasks with a frozen CLIP encoder (Goh et al., 2021), while training a perceiver resampler and cross-attention with multimodal data on a LM objective. To reduce demands on supervised vision-text data, Wang et al. (2021c) pretrained a VL model from scratch using weak-labeled vision and text data. Despite various of parameter-efficient methods are applied during pretraining the models (Li and Liang, 2021; Morrone et al., 2019; Wang et al., 2021b; Kamath et al., 2020), these models are often of over billions of parameters and require high computing machines for deployment.

VL-T5 (Cho et al., 2021) is a moderate-size VL model, where the T5 backbone is updated on multitask objectives, with the encoder jointly learning from Faster-RCNN (Fu et al., 2021) features and input texts. FewVLM (Jin et al., 2021) improved on VL-T5 with prompt-based learning and simplified LM pretraining objectives. Sung et al. (2022) proposed adapter-based fine-tuning on downstream tasks. We exploit the potential of these moderate-size VL models and propose a more parameter-efficient few-shot learner with adapter-based pretraining.

3 Problem Statement

Despite that larger models are significantly more powerful following the scaling laws (Kaplan et al., 2020), we aim to answer the following key questions: i) how can we maximize the efficiency in transferring knowledge of a moderate-size frozen LM to a multimodal setting? ii) how much can we achieve on few-shot learning if we limit the size of data and trainable parameters in VL pretraining?

4 Method

This section describes MAP in details. Our approach is to maximize the knowledge transfer of a moderate-size LM to VL learning through adapter-based pretraining and prompting.

4.1 Model Architecture

We adopt a transformer-based encoder-decoder architecture (Parmar et al., 2017) to jointly encode vision and language inputs and generate target texts. As shown in Figure 2, the model is

Task	Input prompt	Example	
VQA	[Q]	**input:** What is this bird called?	**output:** parrot
	[Q] <text_1>	**input:** What is this bird called? <text_1>	**output:** parrot
	question: [Q] answer:	**input:** question: What is this bird called? answer:	**output:** parrot
	question: [Q] answer: <text_1>	**input:** question: What is this bird called? answer: <text_1>	**output:** parrot
Visual Entailment	[Q]	**input:** A hot air balloon is making a landing.	**output:** entailment
	[Q] <text_1>	**input:** A hot air balloon is making a landing. <text_1>	**output:** entailment
	hypothesis: [Q] label:	**input:** hypothesis: A hot air balloon is making a landing. label:	**output:** entailment
	hypothesis: [Q] label: <text_1>	**input:** hypothesis: A hot air balloon is making a landing. label:<text_1>	**output:** entailment

Table 1: Hand-crafted prompts. For VQA tasks, we prompt with "$question$:" for the input questions with "$answer$:" before the model output. A specific token "<$text\_1$>" is used to indicate the generated words we expect (Jin et al., 2021). Similarly, we designed prompts of "$hypothesis$:" and "$label$:" for VE tasks.

mainly composed of three parts: a visual encoder, a transformer-based encoder-decoder LM backbone, and a series of adapter layers.

Visual Encoder Following Tsimpoukelli et al. (2021), we use a NFResNet encoder (Brock et al., 2021) to convert input images into visual embeddings. The visual embedding vectors then serve as prefixes to be jointly taken with text embeddings by the pretrained language model.

Encoder-decoder LM We adopt a moderate-size pretrained encoder-decoder LM, T5-base (Raffel et al., 2019), as the backbone of the model. The encoder builds joint representation of the input image-text pairs by taking the concatenated visual and text embeddings. Then, the decoder generates target texts in an auto-regressive manner.

Adapters Following Eichenberg et al. (2022), we use the bottleneck adapter modules, which are essential scaled residual bottleneck MLPs (Equation 1). The parameters of the adapters are updated instead of the entire model during pretraining. We add the adapters to the feed-forward and the attention blocks of the transformer following practical analysis by Eichenberg et al. (2022). In the transformer decoder, we experiment with different settings of adding the adapter layers to cross-attention or self-attention blocks.

$$A(h) = h + \lambda W^{up}\phi(W^{down}h) \tag{1}$$

4.2 Pretraining

Following Jin et al. (2021), we pretrain MAP on MaskedLM (Chang et al., 2018) and PrefixLM with paired image-caption data (Liu et al., 2019). However, instead of updating the entire parameter set of the LM, we only update the parameters of the adapter layers. We experiment with both settings of updating or freezing the visual encoder. Our adapter-based end-to-end pretraining is illustrated in Figure 2.

4.3 Few-shot Learning

In downstream tasks, we experiment with few-shot learning with both prompting and in-context learning. For prompting, we use hand-crafted prompts (Jin et al., 2021) and train the model with few-shot examples to minimize the negative-log-likelihood (Table 1). For in-context learning, we concatenate a series of image-text pairs in order as a multimodal prompt and expect the model to predict the target text given a visual query.

5 Experiments

5.1 Datasets

For pretraining, we combine image-caption pairs from MS COCO caption (Zitnick et al., 2015) and Visual Genome (VG) (Bernstein et al., 2017).[*] To explore the influence of different pretrained data size, we designed 3 versions of the pretraining data, with the corresponding number of VG region-caption pairs extracted from each image set to 2, 10 and 36. This leads to 0.3M, 0.8M and 4.2M image-caption pairs. We do not include any downstream dataset in pretraining.

We evaluate MAP's transfer ability on five downstream tasks, including VQAv2, OKVQA, GQA, and VizWiz for visual question answering, and SNLI-VE for language-image understanding.

5.2 Training Details

Training Settings For pretraining, we set batch size as 240 and pretrain with 30 epochs. We use learning rate 1e-4 with 5% linear warmup. We build the model with PyTorch and run on 8 A100 GPUs for around 5 days. For few-shot learning, we use learning rate 5e-5 with 5% linear warmup and

[*] As the annotated captions in Visual Genome are region descriptions, MAP directly takes the image drawn with the corresponding bounding boxes.

[†] We report the accuracy with MAGMA model using the NFResNet encoder.

[‡] 592K(COCO)+36*108K(VG)

Method	$\|Data\|$	VQAv2	OK-VQA	GQA	SNLI-VE	VizWiz	$\|LM\|$
VL-T5$_{novqa}$ (Cho et al., 2021)	4.9M	31.8	12.7	19.6	-	-	224M
VL-T5$_{novqa}$ (Cho et al., 2021)	0.3M	0.1	0.0	0.0	-	-	224M
FEWVLM$_{base}$ (Jin et al., 2021)	4.9M	**48.2**	15.0	**32.2**	-	-	224M
FEWVLM$_{base}$ (Jin et al., 2021)	0.3M	16.8	9.9	13.2	-	-	224M
MAP$_{base}$	4.2M	40.4	**17.1**	27.2	33.1	25.6	224M*
MAP$_{small}$	0.8M	40.5	16.8	22.9	32.5	25.2	224M*
MAP$_{tiny}$	0.3M	38.0	15.7	22.1	**41.9**	**27.9**	224M*

Table 2: Few-shot (16-shot) evaluation results on VQAv2, OK-VQA, GQA, VizWiz and SNLI-VE. Compared to baseline models, MAP can be trained with much fewer pre-training data and parameters with minor downstream performance degradation. The * symbol indicates the parameters are frozen during pretraining.

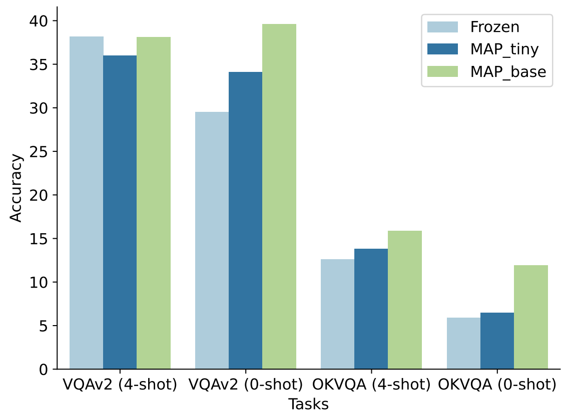


Figure 3: MAP outperforms Frozen on zero-shot VQAv2, zero-shot OK-VQA, and four-shot OK-VQA with a 30× lighter LM backbone (a 224M T5-base compared to a 7B GPT-style LM). Gains are retained even when using only 10% the size of pretraining data (0.3M for MAP$_{tiny}$ compared to 3.0M for Frozen.)

train for 200 epochs with the size of 16 for D_{train} and D_{dev}. We choose the best checkpoint for test set evaluation.

Hand-crafted Prompts As shown in Table 1, we use task-specific prompts designed for downstream evaluations to make the most of the transfer ability from the pre-trained model. We experiment of three different templates with corresponding input and result hints for VQA and VE tasks. Our prompts for VQA follows the design by Jin et al. (2021).

6 Evaluation and Results

To answer the questions that we raised in Section 3, we evaluate MAP on the aforementioned five downstream tasks in zero-shot and few-shot settings.

From our preliminary experiments, jointly updating the NFResNet vision encoder and adapter layers performs slightly better in pretraining than updating adapters only. We therefore applied such settings in all models pretrained using COCO combined with three versions of VG region-caption pairs, denoting as MAP$_{tiny}$, MAP$_{small}$ and MAP$_{base}$.

Transfer Efficiency To evaluate MAP's efficiency on transferring knowledge from a **frozen** LM to a multimodal setting, we compare MAP against Frozen on VQAv2 and OK-VQA. As shown in Figure 3, overall, MAP achieves better zero-shot and few-shot performance on both tasks. MAP$_{tiny}$ is able to outperform Frozen on zero-shot VQAv2, zero-shot and four-shot OK-VQA even with only 10% the size of pretraining data (0.3M v.s. 3.0M) and a 30× lighter LM backbone (224M v.s. 7B).

Data and Parameter Efficiency We compare MAP to fully trained VL models to evaluate how much can be achieved with limited pretraining data and number of trainable parameters. In Table 2, we show that compared to VL-T5 (Cho et al., 2021), on all the five downstream tasks, MAP$_{tiny}$ achieves much better results with only 16% in size of the pre-training dataset[†] and 48% in the number of trainable parameters. Moreover, MAP is strong at retaining its transfer learning ability while VL-T5 and FewVLM adapt the language modeling ability to the 0.3M pretraining data.

7 Conclusion and Future Work

We present an end-to-end moderate-size VL model, which surpasses Frozen and comparable-size fully trained baselines on few-shot learning over multiple image understanding tasks, while requiring much less training data and fewer parameters during pretraining. We expect to investigate the core transfer ability of pretrained VL models from a perspective beyond scaling. We propose open questions of whether data can be overloaded in pretraining and how can we use pretraining data more efficiently and wisely.

[†]Here we consider only the data on the LM objective. VL-T5 uses additional 3.3M multi-task data in pretraining.

Limitations

We experiment with two concatenation methods to build sequential VL inputs for in-context learning. However, we do not see improvements in few-shot performance with either settings, which may root in its pretraining strategy of taking only single image-caption pairs. Details are illustrated in Appendix A.

While our pretrained model obtains strong few-shot learning ability through parameter-efficient pretraining with a much smaller dataset, it is also possible that the small number of trainable parameters could limit its ability to learn from large-scale dataset. It is still an open question of how to automatically select multimodal data samples and maximizing data efficiency during the learning process.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. Flamingo: a visual language model for few-shot learning.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, Tadas Baltrusaitis, Amir Zadeh, Yao Lim, Ankur Bapna, Yu-An Chung, Nan Wu, Anmol Gulati, Andrew Brock, Soham De, Samuel Smith, Tom Brown, Benjamin Mann, Nick Ryder, Jared Subbiah, Prafulla Kaplan, Arvind Dhariwal, Pranav Neelakantan, Girish Shyam, Amanda Sastry, Sandhini Askell, and Ariel Agarwal. 2022. Prompting as multimodal fusing.

Michael S. Bernstein, Yuke Zhu, Oliver Groth, Joshua Kravitz, Ranjay Krishna, Kenji Hata, Li-Jia Li, David A. Shamma, Yannis Kalantidis, Justin Johnson, Li Fei-Fei, and Stephanie Chen. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*.

Andrew Brock, Soham De, Samuel L. Smith, and Karen Simonyan. 2021. High-performance large-scale image recognition without normalization. *international conference on machine learning*.

Ming-Wei Chang, Kenton Lee, Kristina Toutanova, and Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *north american chapter of the association for computational linguistics*.

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. *arXiv: Computation and Language*.

Constantin Eichenberg, Sidney Black, Samuel Weinbach, Letitia Parcalabescu, and Anette Frank. 2022. Magma – multimodal augmentation of generative models through adapter-based finetuning.

Jianlong Fu, Dongmei Fu, Bei Liu, Zhicheng Huang, Zhaoyang Zeng, and Yupan Huang. 2021. Seeing out of the box: End-to-end pre-training for vision-language representation learning. *computer vision and pattern recognition*.

Gabriel Goh, Girish Sastry, Pamela Mishkin, Jong Wook Kim, Alec Radford, Sandhini Agarwal, Chris Hallacy, Jack Clark, Amanda Askell, Gretchen Krueger, Ilya Sutskever, and Aditya Ramesh. 2021. Learning transferable visual models from natural language supervision. *international conference on machine learning*.

Woojeong Jin, Yu Cheng, Yelong Shen, Weizhu Chen, and Xiang Ren. 2021. A good prompt is worth millions of parameters? low-resource prompt-based learning for vision-language models. *arXiv: Computer Vision and Pattern Recognition*.

Aishwarya Kamath, Sebastian Ruder, Clifton Poth, Andreas Rücklé, Jonas Pfeiffer, Iryna Gurevych, Ivan Vulić, and Kyunghyun Cho. 2020. Adapterhub: A framework for adapting transformers. *empirical methods in natural language processing*.

Jared Kaplan, Samuel McCandlish, Thomas Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv: Learning*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *meeting of the association for computational linguistics*.

Peter J. Liu, Michael Matena, Katherine Lee, Adam Roberts, Yanqi Zhou, Noam Shazeer, Colin Raffel, Sharan Narang, and Wei Li. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Ziyang Luo, Yadong Xi, Rongsheng Zhang, and Jing Ma. 2022. I-tuning: Tuning language models with image for caption generation.

Bruna Halila Morrone, Neil Houlsby, Sylvain Gelly, Mona Attariyan, Stanisław Jastrzębski, Andrei Giurgiu, Andrea Gesmundo, and Quentin de Laroussilhe. 2019. Parameter-efficient transfer learning for nlp. *international conference on machine learning*.

Niki Parmar, Lukasz Kaiser, Jakob Uszkoreit, Illia Polosukhin, Aidan N. Gomez, Ashish Vaswani, Noam Shazeer, and Llion Jones. 2017. Attention is all you need. *neural information processing systems*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. 2022. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks.

Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. *arXiv: Computer Vision and Pattern Recognition*.

Lijuan Wang, Zhe Gan, Xiaowei Hu, Yumao Lu, Zhengyuan Yang, Jianfeng Wang, and Zicheng Liu. 2021a. An empirical study of gpt-3 for few-shot knowledge-based vqa. *arXiv: Computer Vision and Pattern Recognition*.

Shean Wang, Yelong Shen, Zeyuan Allen-Zhu, Yuanzhi Li, Phillip Wallis, Weizhu Chen, and Edward J. Hu. 2021b. Lora: Low-rank adaptation of large language models. *arXiv: Computation and Language*.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2021c. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv: Computer Vision and Pattern Recognition*.

C. Lawrence Zitnick, Tsung-Yi Lin, Xinlei Chen, Saurabh Gupta, Ramakrishna Vedantam, Piotr Dollár, and Hao Fang. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv: Computer Vision and Pattern Recognition*.

A In-context Learning

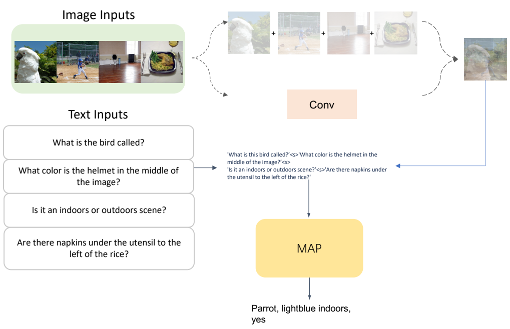


Figure 4: Concatenation Illustration

Our two approaches in concatenating inputs are illustrated in Figure 4. One is to mix-up images obtained by multiplying averaged weights within one glance and adding them all together with normalization, and the other way is to concatenate them on the channel dimension and pass through a convolution layer before feeding into the visual encoder.

B Modality Fusion

To validate that our joint-encoder's ability in learning multimodal representations, we apply linear probing on the representation output by the encoder and reach 66.4% in accuracy on the SNLI-VE task.

On the role of resources in the age of large language models

Simon Dobnik
FLoV and CLASP
University of Gothenburg, Sweden
simon.dobnik@gu.se

John D Kelleher
Hamilton Institute
Maynooth University, Ireland
john.kelleher@mu.ie

Abstract

We evaluate the role of expert-based domain knowledge and resources in relation to training large language models by referring to our work on training and evaluating neural models, also in under-resourced scenarios which we believe also informs training models for "well-resourced" languages and domains. We argue that our community needs both large-scale datasets and small but high-quality data based on expert knowledge and that both activities should work hand-in-hand.

1 Introduction

In the recent years large language models based on transformers that are trained end to end and automatically capture the structure of language have achieved remarkable performance (Devlin et al., 2018; Brown et al., 2020). Indeed there is an ongoing debate as to whether level of semantics that these systems obtain. On the one hand, some researchers argue that systems trained on linguistic form alone are limited to being "statistical parrots" (Bender and Koller, 2020) and others argue the correspondence between language use and situations in the world enables these systems to access meaning (Sahlgren and Carlsson, 2021).[1]. Alongside the debate in relation to the semantics these systems encode, several questions have been raised in relation to their training and usage.

Large language models require a lot of data to train and to do that an approach in natural language processing has been to utilise (sometime indiscriminately) all the data that is available. However, access to the data is heavily biased to the data that can be found online, e.g. Wikipedia, or data that can be collected with crowd-sourcing platforms. Such selection of data on which the models are trained does not represent all possible contexts of language use or groups of society producing language which results in undesired and exaggerated thematic (Agrawal et al., 2017) and social bias (Bender et al., 2021) in the models. Moreover, although the performance improvements of LLMs across a range of tasks has come in tandem with a massive growth in the dataset and model sizes and the compute used to train these systems (see, e.g. (Kaplan et al., 2020)), it is recognised that quality data is core to these improvements and there are some projections based on the current rate of growth in data requirements that we may run out of quality training data in the near future (Villalobos et al., 2022). Indeed, a response to this challenge can be seen in the significant amount of current research focused on how to automatically curate quality data from huge web crawl datasets (Penedo et al., 2023).

In addition to access to large datasets of text (and images), training such models is also costly in terms of time and available computational resources, both factors which are only available to a few world languages where English is over-represented.

On the other hand, curating of datasets in terms of collecting high-quality data and their annotation with linguistically-motivated annotation schemes has a long tradition in natural language processing. Transformer models learn linguistic structure end-to-end and systems using automatically learned contextualised embedding surpass models with expert-engineered features which raises a question whether all the years of hard expert work is superfluous. But can we be really sure that the models really have learned useful linguistic structure (Conneau et al., 2018)? Is that structure the same what we expect (Dobnik et al., 2018)? Since annotation of resources is directly connected with linguistics, which focuses on understanding of differences between languages and therefore explores a variety of world languages, the annotation work provides a good cross-linguistic coverage but

[1]See (Kelleher and Dobnik, 2022) for more on this debate.

Proceedings of the 2023 CLASP Conference on Learning with Small Data, pages 191–197
September 11–12, 2023. ©2023 Association for Computational Linguistics

frequently datasets have a limited coverage of examples and may not be large enough for training machine learning models. Another benefit of a close relation of this approach to linguistics is that the annotation categories are motivated by our (expert) understanding of how these languages work so the resulting representations are well-motivated and interpretable.

In this presentation we evaluate the previous questions about the role of data and resources for modern natural language processing in the light of our experience with building resources for under-resourced language from ground up. We highlight the idea that in such scenarios both kinds of resources are useful and in fact shows that they have complementary weaknesses and strengths. It follows that modern and future natural language processing must be informed by expert domain knowledge about language and linguistics as without these we are not able to evaluate the data that these models are utilising nor interpret what semantics or bias the models might have captured nor we can improve the models in a motivated way either indirectly (by neural architecture choice) or directly (by injection of labels).

2 The need for (deeper) semantics

One of the primary objective functions used for training large foundational language models is to predict the next word or a missing word from the surrounding context. This objective function indirectly priorities several linguistic perspectives. The of semantics that is learned in this way can be characterised as being primarily *distributional* (in the sense of (Firth, 1957)), *thematic* (rather than *taxonomic*, see (Kacmajor and Kelleher, 2020)), and *topical* (in the sense of topic as word co-occurrence (Manning and Schutze, 1999)).

Given that the semantics of these models is primarily based on co-occurrence an interesting question to ask is whether (or at least how far) can co-occurrence bring a model in terms of semantics. A review of the literature probing on neural embeddings (Conneau et al., 2018) and on the BERT architecture in particular (Rogers et al., 2020) indicates that neural embeddings do encode a range of linguistic information, in particular topic and syntax. However, a relatively under-explored aspect of these systems is their ability to capture and encode semantic phenomena, such as idiomaticity (Nedumpozhimana et al., 2022). One reason for this lack of research is the relative paucity of large scale annotated benchmarks for semantic phenomena. For example, many probing experiments build on the benchmark datasets set out in (Conneau et al., 2018), however the tasks covered by these datasets are primarily syntactic in nature (Klubicka and Kelleher, 2022). There is some evidence that BERT does encode semantic phenomena (Nedumpozhimana and Kelleher, 2021). However, in the current context of large language models trained on massive datasets the question of whether *more is different* is holds for linguistic semantics arises?

Some researchers have argued that BERT rediscovers the classical NLP pipeline, with the earlier layers encoding syntactic information and later layers semantic (Tenney et al., 2019). However, a number of recent studies have found that the performance of BERT-based transformer models across a range of standard NLP benchmark dataset is robust to word-order perturbations (see e.g. (Pham et al., 2021), (Gupta et al., 2021)). These results suggest transformer based models such as BERT rely on relatively shallow surface level information such as topic rather than syntactic information. Moreover, this suggests current NLP benchmarks are not challenging enough to comprehensively assess linguistic semantic (Sinha et al., 2021). The difficulties in developing robust benchmark datasets has been raised in the discussions around the recent work by (Jiang et al., 2023) that reported a set of experiments that demonstrate that a simple *gzip*-based text classification method outperforms BERT and a number of other deep neural network NLP models on a range of text classification tasks using standard datasets.

3 The need for data

Training of large language models requires a lot of data that spans over different contexts of language use and social groups in order to capture (some kind of) knowledge of language for natural language generation and interpretation and to avoid unwanted social and contextual bias. However, as discussed in the previous section even for well-resourced language models such as English it is still not clear whether this has been achieved as data selection and coverage of thematic and social contexts that are used in the training data has not yet been (to our knowledge) systematically evaluated. Equally, approaching the same problem from the engineering perspective it has been impossible to collect enough data or build a model large enough to test whether such an endeavour is theoretically and practically possible at all (Villalobos et al., 2022).

This need for data and its limitations becomes much more evident when we examine the under-resourced scenarios that we looked at. Arabic natural language processing is an interesting case. Modern Standard Arabic (MSA) is a standardised form of Arabic used in printed media and news and is supported well in terms of natural language models and resources. However, there are also several local varieties spoken over a large geographical span. In addition, Arabic may be also spoken (and written in social media) and code-switched with several other varieties and even different languages. Some of these have received more attention than others in NLP. For example, there has been a good support for the Egyptian variety but very little support for Algerian and the individual varieties in the Levantine area. Another interesting aspect of Arabic linguistic landscape is that it differs between regions/countries in what situation contexts different varieties are used, what other varieties are present in these contexts and how similar these varieties are.

Speakers/writers in Algeria (Adouane and Dobnik, 2017) use social media where varieties that were typically spoken in personal everyday communication are now written with Arabic script on a limited phone keyboard. There is no standard spelling for these varieties and the practical limitation of using different keyboards introduce high level of variation in the way these varieties are written by different users in different contexts on different social media. A further level of of variation is added when these varieties are code-switched with MSA and other languages, in case of Algerian with Berber, French and English, all written in the same script. Hence, one of the first tasks to tackle the bootstrapping of resources for Algerian was to build a code-switching detector based on a limited expert-annotated corpus using probabilistic (HMM) and bi-gram feature classification models.

On the other hand, Levantine dialects (Abu Kwaik et al., 2018b) are various closely related Arabic dialects that are spoken and written in social media but such context makes them hard to distinguish from each other as phonological form which underlies a lot of discriminating power is missing (Abu Kwaik et al., 2018a). Finally, Wolaytta (Gebreselassie and Dobnik, 2022), is one of several languages spoken in Ethiopia, belongs to the Omotic family of African languages which is different from Amharic, an official national language which belongs to the Semitic family of languages and for which most NLP resources exits. Wolaytta is mostly used in spoken form in personal communication and radio and has been standardised in the written form in school texts and religious textbooks. In terms of NLP resources, there is no social media but they are radio programmes , school textbooks, religious literature and a Wolaytta-English dictionary.

Comparing these cases we can see that there are large linguistic differences between these target varieties and the language used in the closest set of contexts for which NLP resources exist and also that we have limited records of of contexts in which they are used either be cause data is missing or because the variety is not used in those contexts. Consequently, building NLP resources had to rely on a large support from expert linguistic and social knowledge because the training examples were limited we relied on simple machine learning methods such as Bayesian classification which in conjunction with the expert knowledge gave satisfactory results.

4 The need for the right method

Different (i) contexts of language use, (ii) relation to the closest variety for which NLP resources exits, (iii) availability of data, (iv) availability of expert annotation required very different tools and approaches to build resources and NLP applications for these varieties. For example, using character and sub-word models and CNNs, weak supervision (bootstrapping from an existing labeller, self-training) (Adouane et al., 2018b; Abu Kwaik et al., 2020), injecting background knowledge from lexicon and pre-trained sub-word embeddings (Adouane et al., 2018a), pre-training (Abu Kwaik et al., 2022), text normalisation with alignment of tokens (Adouane et al., 2019b), data augmentation (Adouane et al., 2019a). It is often the case that a simple model works better than a more complex model, most likely because it is able to generalise better from a limited data (Abu Kwaik et al., 2019a,b). In sum, understanding language and its context is important even at the age of large language models to make an informed choice what model should be used when.

5 Are "well-resourced" languages also under-resourced?

We argued previously that there is still an open question whether language model have in fact reached understanding of language as they have not been exposed to all contexts of language use. Hence, we are facing with similar under-resourced scenarios also in cases of "well-resourced" languages where existing large language models are applied in contexts or tasks for which the model has not been initially trained on. Language is continuously changing and speakers/writers are creative, especially in social media (Noble et al., 2021). Hence, pre-trained language models may become quickly outdated.

Our work on generating spatial descriptions of images shows that since pre-training of visual features such as ResNet (He et al., 2016), Faster-RCNN (Ren et al., 2015) and CLIP (Radford et al., 2021) that are trained to identify objects affects what is model able to learn about predicting relations which are likely to be hallucinated from a language model, simply because the model has not been pre-trained in this way and until such features are explicitly identified (Ghanimifard and Dobnik, 2019). A significant body of work on language and vision has focused on generation of image descriptions that focus on a single sentence. Extending the task to multi-sentence generation requires application of different models (Ilinykh and Dobnik, 2020).

Adaption of pre-trained models from the image captioning domain on object classification (where objects are in the attention focus of the scene) to the domain of situated language (where a robot without a specific model of visual and thematic attention) is very different reveals that visual information in such cases is used quite differently than in an image captioning scenario (Ilinykh et al., 2022).

Finally, a comparison of generated noun phrases in generated multi-sentence descriptions to human descriptions (Ilinykh and Dobnik, 2022) reveals a difference. Models are more general predictors than humans across the board and opt for more general descriptions of objects than humans. This is because they are trained on a single task, but also within this task they are biased to find a single generalisation following a training objective covering all of the examples equally, whereas in reality humans might use descriptions that are more general or more specific on a case-to-case bases. Since general descriptions are more frequent than the specific ones, they always win. Overall, it appears that a very fine grained knowledge of language data is required to capture all the contexts.

6 Conclusions

Training language and vision and language models is useful but so is production of high quality domain specific resources as both tasks are complementary. We might want to rethink how to train such models – having one large model is practical, but perhaps not the end of the NLP story and more work is required to examine the limits of models to capture a variety of possible contexts. Expert-based knowledge is highly relevant for selecting content data, creating datasets, and evaluating contexts in which models are trained. Similarly, expert-based resources are relevant to make informed choices about the model architectures and to support training of end-to-end models by feature engineering and selection. This also includes application of pre-trained feature representations. Understanding architectures, models and training regimes allows us to define the limits of what linguistic knowledge can be represented and learned and should inform data preparation and annotation work. Although significant work has been done on evaluating the models for acquired linguistic knowledge, more targeted fine-grained evaluation of models is necessary to achieve the models fit to the previous requirements, with targeted positive and negative linguistic examples (beyond the level of granularity of a Turing test as implemented in the GLUE benchmarks (Wang et al., 2019)), which is one of our current efforts.

Acknowledgements

The research reported in this paper was supported by a grant from the Swedish Research Council (VR project 2014-39) for the establishment of the Centre for Linguistic Theory and Studies in Probability (CLASP) at the University of Gothenburg.

References

Kathrein Abu Kwaik, Stergios Chatzikyriakidis, and Simon Dobnik. 2019a. Can modern standard Arabic approaches be used for Arabic dialects? Sentiment analysis as a case study. In *Proceedings of the 3rd Workshop on Arabic Corpus Linguistics (WACL-3)*, pages 40–50, Cardiff, United Kingdom. Association for Computational Linguistics.

Kathrein Abu Kwaik, Stergios Chatzikyriakidis, and Simon Dobnik. 2022. Pre-trained models or feature engineering: The case of dialectal Arabic. In *Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection (OSACT) at LREC 2022*, pages 41–50, Marseille, France. European Language Resources Association.

Kathrein Abu Kwaik, Stergios Chatzikyriakidis, Simon Dobnik, Motaz Saad, and Richard Johansson. 2020. An Arabic tweets sentiment analysis dataset (ATSAD) using distant supervision and self training. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools with a Shared Task on Offensive Language Detection (OSACT4-2020) at Language Resources and Evaluation Conference (LREC 2020)*, pages 1–8, Marseille, France. European Language Resources Association (ELRA).

Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnik. 2018a. A lexical distance study of Arabic dialects. *Procedia Computer Science 142: Proceedings of the 4th International Conference on Arabic Computational Linguistics (ACLing 2018)*, 142:2–13.

Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnik. 2018b. Shami: a corpus of Levantine Arabic dialects. In *Proceedings of LREC 2018, 11th International Conference on Language Resources and Evaluation*, pages 1–8, Phoenix Seagaia Conference Center, Miyazaki, Japan. European Language Resources Association (ELRA).

Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnik. 2019b. LSTM-CNN deep learning model for sentiment analysis of dialectal Arabic. In *Proceedings of ICALP'19: The 7th International Conference on Arabic Language Processing*, Communications in Computer and Information Science (CCIS), pages 1–14, Nancy, France. Springer.

Wafia Adouane, Jean-Philippe Bernardy, and Simon Dobnik. 2018a. Improving neural network performance by injecting background knowledge: Detecting code-switching and borrowing in Algerian texts. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching at 56th Annual Meeting of the Association for Computational Linguistics (ACL-2018)*, pages 20–28, Melbourne, Australia. Association for Computational Linguistics.

Wafia Adouane, Jean-Philippe Bernardy, and Simon Dobnik. 2019a. Neural models for detecting binary semantic textual similarity for Algerian and MSA. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop WANLP 2019 at ACL-2019*, pages 78–87, Florence, Italy. Association for Computational Linguistics.

Wafia Adouane, Jean-Philippe Bernardy, and Simon Dobnik. 2019b. Normalising non-standardised orthography in Algerian code-switched user-generated data. In *Proceedings of The 5th Workshop on Noisy User-generated Text (W-NUT) at EMNLP 2019*, pages 1–10, Hong Kong. Ritter, Alan and Xu, Wei and Baldwin, Tim and Rahimi, Afshin.

Wafia Adouane and Simon Dobnik. 2017. Identification of languages in Algerian Arabic multilingual documents. In *Proceedings of The Third Arabic Natural Language Processing Workshop (WANLP)*, pages 1–8, Valenica, Spain. The European Chapter of the Association for Computational Linguistics (EACL), Association for Computational Linguistics.

Wafia Adouane, Simon Dobnik, Jean-Philippe Bernardy, and Nasredine Semmar. 2018b. A comparison of character neural language model and bootstrapping for language identification in multilingual noisy texts. In *Proceedings of the Second Workshop on Subword and Character Level Models in NLP (SCLeM) at 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)*, pages 1–10, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2017. Don't just assume; look and answer: Overcoming priors for visual question answering. *arXiv*, arXiv:1712.00377 [cs.CV]:1–15.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pages 610–623, New York, NY, USA. Association for Computing Machinery.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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. *arXiv*, arXiv:2005.14165 [cs.CL]:1–75.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv*, arXiv:1805.01070 [cs.CL].

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *arXiv*, arXiv:1810.04805 [cs.CL]:1–14.

Simon Dobnik, Mehdi Ghanimifard, and John D. Kelleher. 2018. Exploring the functional and geometric bias of spatial relations using neural language models. In *Proceedings of the First International Workshop on Spatial Language Understanding (SpLU 2018) at NAACL-HLT 2018*, pages 1–11, New Orleans, Louisiana, USA. Association for Computational Linguistics.

John Firth. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, pages 10–32.

Tewodros Gebreselassie and Simon Dobnik. 2022. Wolaytta word embeddings. Technical report, manuscript, Centre for Linguistic Theory and Studies in Probability (CLASP), Gothenburg, Sweden.

Mehdi Ghanimifard and Simon Dobnik. 2019. What goes into a word: generating image descriptions with top-down spatial knowledge. In *Proceedings of the 12th International Conference on Natural Language Generation (INLG-2019)*, pages 1–15, Tokyo, Japan. Association for Computational Linguistics.

Ashim Gupta, Giorgi Kvernadze, and Vivek Srikumar. 2021. Bert & family eat word salad: Experiments with text understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12946–12954.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Nikolai Ilinykh and Simon Dobnik. 2020. When an image tells a story: The role of visual and semantic information for generating paragraph descriptions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 338–348, Dublin, Ireland. Association for Computational Linguistics.

Nikolai Ilinykh and Simon Dobnik. 2022. Do decoding algorithms capture discourse structure in multi-modal tasks? A case study of image paragraph generation. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 480–493, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nikolai Ilinykh, Yasmeen Emampoor, and Simon Dobnik. 2022. Look and answer the question: On the role of vision in embodied question answering. In *Proceedings of the 15th International Conference on Natural Language Generation (INLG)*, Colby College, Waterville, ME, USA.

Zhiying Jiang, Matthew Yang, Mikhail Tsirlin, Raphael Tang, Yiqin Dai, and Jimmy Lin. 2023. "low-resource" text classification: A parameter-free classification method with compressors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6810–6828, Toronto, Canada. Association for Computational Linguistics.

Magdalena Kacmajor and John D Kelleher. 2020. Capturing and measuring thematic relatedness. *Language Resources and Evaluation*, 54(3):645–682.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

John D. Kelleher and Simon Dobnik. 2022. Distributional semantics for situated spatial language? functional, geometric and perceptual perspectives. In *Probabilistic Approaches to Linguistic Theory*. CSLI Publications.

Filip Klubicka and John Kelleher. 2022. Probing with noise: Unpicking the warp and weft of embeddings. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 404–417, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Christopher Manning and Hinrich Schutze. 1999. *Foundations of statistical natural language processing*. MIT press.

Vasudevan Nedumpozhimana and John Kelleher. 2021. Finding BERT's idiomatic key. In *Proceedings of the 17th Workshop on Multiword Expressions (MWE 2021)*, pages 57–62.

Vasudevan Nedumpozhimana, Filip Klubička, and John D. Kelleher. 2022. Shapley idioms: Analysing BERT sentence embeddings for general idiom token identification. *Frontiers in Artificial Intelligence*, 5.

Bill Noble, Asad Sayeed, Raquel Fernández, and Staffan Larsson. 2021. Semantic shift in social networks. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 26–37, Online. Association for Computational Linguistics.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Thang Pham, Trung Bui, Long Mai, and Anh Nguyen. 2021. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1145–1160, Online. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Magnus Sahlgren and Fredrik Carlsson. 2021. The singleton fallacy: Why current critiques of language models miss the point. *Frontiers in Artificial Intelligence*, 4:131.

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.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Pablo Villalobos, Jaime Sevilla, Lennart Heim, Tamay Besiroglu, Marius Hobbhahn, and Anson Ho. 2022. Will we run out of data? an analysis of the limits of scaling datasets in machine learning. *arXiv*, arXiv:2211.04325 [cs.LG].

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Author Index