Research Master Thesis

Cross-lingual Transfer Using Stacked Language Adapters

Marcell Fekete

a thesis submitted in partial fulfilment of the requirements for the degree of

MA Linguistics
(Human Language Technology)

Vrije Universiteit Amsterdam

Computational Lexicology and Terminology Lab
Department of Language and Communication
Faculty of Humanities

Supervised by: Dr. Lisa Beinborn
2^{nd} reader: Dr. Pia Sommerauer

Submitted: July 1, 2022

Abstract

The world's languages are far from equal in terms of representation in the online space. While extant research in natural language processing (NLP) delivers impressive tools for on languages with vast amounts of data available, the majority of languages do not have the same amount of data available. This means that with current methods in NLP, billions of people may never have access to these tools in their native language. Recent attempts address this issue by exploring *cross-lingual transfer*, the transfer of information between languages inside language models. This phenomenon seems to be facilitated by shared linguistic and extra-linguistic properties between languages.

In this thesis, I contribute to this research direction. I set out to evaluate how cross-lingual transfer might be influenced by the similarity in morphological complexity between languages, a specific typological feature that language models were shown to be sensitive to. For this, I use a technique called language adapters, a technique that allows efficient and flexible language model adaptation. By stacking (combining) language adapters pretrained on different languages, I assess whether cross-lingual transfer can be induced between these languages. I investigate whether language model performance can be improved on the low resource languages, and whether this transfer is more successful when the languages are similar to each other in terms of morphological complexity. With these goals in mind, I carry out 80 experiments on a diverse set of 8 languages using Tatoeba, a *cross-lingual sentence retrieval* (translation pair detection).

I find that stacked language adapters can contribute to better performance for low resource languages on the evaluation task, demonstrating successful cross-lingual transfer between the adapters. However, the success of this transfer does not appear to depend on similarity between languages in terms of morphological complexity. Other linguistic and extra-linguistic factors – including pretraining data size of the different languages and shared script – are assessed, but further investigation is necessary to reveal which factors condition positive cross-lingual transfer between stacked language adapters.

Declaration of Authorship

I, Marcell Richard Fekete, declare that this thesis, titled *Cross-lingual Transfer Using Stacked Language Adapters* and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a degree degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: 30th June, 2022

Signed: 

Acknowledgments

I am grateful for the help of my supervisor, Lisa Beinborn, and my colleagues at TAUS. Special thanks for the researchers that contributed with providing access to important datasets: Sabrina J. Mielke and Jack Rueter. I would also like to thank Wondimagegnhue (Wende) Tsegaye Tufa for our reading groups that shed added to my understanding of related work, and Jose Angel Daza Arevalo for his guidance on the research direction.

I would also like to thank my friends and my family, and of course Shahrin. Without their support, this thesis would not exist.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Language models drive many of our interactions with technology. Search engines are essential in retrieving information, neural machine translation allows communication across languages, and voice assistants provide not only convenience, but also accessibility. These are just a few examples in how natural language processing (NLP) facilitates access to many tools our society relies on.

Recent years brought impressive developments in the field. But while there are approximately 7,000 languages spoken in the world, native access to this technology is limited to a small subset. The primary reason behind this is that state-of-the-art methods, like neural language models, require vast amounts of data. This is simply not available for the majority of the world's languages. Without addressing this gap between high and low resource languages, billions of people will not have access to the infrastructure that speakers of languages such as English or French can easily use. This can deprive people from convenience and accessibility, or it can force them to switch to a more privileged language to be able to use the same infrastructure speakers of high resource languages have, further marginalising their low resource languages.

Performance differences on different NLP tasks can be considerable between high and low resource languages when using state-of-the-art neural language models, such as XLM-RoBERTa. As a demonstration, let us consider the task of named entity recognition: the goal of the task is to identify named entities in a given text. It is typically measured in F1-score, the harmonic mean of precision (how many words that are identified are actually named entities) and recall (how many of the named entities are correctly identified). On high resource languages, XLM-RoBERTa achieves F1-scores ranging around 80: 84.6 for English, 79.1 for French, and 78.0 for German. Performance, however is considerably worse for low resource languages: it is 49.9 for Kazakh and 41.8 for Yoruba (Ruder et al., 2021).

Many NLP researchers realised that this imbalance between high and low resource languages can only be addressed by considering the individual linguistic and extra-linguistic properties of various languages (Bender, 2009). Linguistic properties refer to typological properties, ones that are inherent to language, such as phonology and morphosyntax, while extra-linguistic properties are external to the properties of language, such as the amount of training data available. Typologically-informed approaches have been proposed to investigate the interaction of structural properties of languages and the capabilities of the language models (Bender, 2011; Ponti et al., 2019; Joshi et al., 2020).

Taking into account properties of the individual languages can inform us on which

languages are more challenging to model regardless of the amount of training data available (Cotterell et al., 2018; Mielke et al., 2019; Park et al., 2021; Jones et al., 2021). There is also research that investigates how information within language models can be transferred between languages through a mechanism of cross-lingual transfer: the sharing of information between different languages inside a model (Pires et al., 2019; Wu and Dredze, 2019; Lin et al., 2019; Bjerva and Augenstein, 2021). These approaches are promising in enabling better performance for languages that lack the massive quantities of data necessary to train state-of-the-art models for.

There is also an angle of *interpretability*: by understanding which linguistic and extra-linguistic properties of languages pose difficulties to current state-of-the-art methods, we can gain an insight into how these language models might process and contextualise information about the different languages.

In my thesis, I set out to investigate how structural similarity between languages, in this case, the complexity of morphological complexity, can facilitate the transfer of model knowledge between different languages. For this, I combine language adapters, flexible and efficient model adaptation modules that are pretrained on specific languages. This thesis can be viewed as an introduction to a general methodology that allows analysing the interaction of various linguistic and extra-linguistic factors using cross-lingual transfer induced between stacked language adapters.

I believe that my main contribution is showing that language adapter stacks can benefit performance on low resource languages, even ones that the language model has not encountered during its training. This shows that it is viable to use stacked language adapters to address the digital gap between high and low resource languages. Since stacking language adapters is easy to carry out, this finding can benefit future research as well as use in industry.

1.1 Research objectives

As I outlined in the previous section, the main motivation of my thesis is making language technology more equitable by extending the number of languages it covers. Concretely, my goal is to enhance language model knowledge especially on low resource languages through exploiting cross-lingual transfer of language model knowledge between various languages. The success of this would result in increased model performance even on languages that previously lacked adequate NLP tools. This would allow researchers and companies to employ these tools that are getting more and more commonplace for high resource languages to new languages.

I carried out my thesis internship with TAUS, a company that specializes in curating parallel data for improving existing neural machine translation methods.[1] For TAUS, the value of this research is that increased language model knowledge on a wider range of languages allows a better performing data processing pipeline. This makes it possible to create higher quality parallel datasets even for low resource languages that can further increase the success of neural machine translation customization and improvement.

In order to address the goal of my thesis, I combine language adapters for different languages. Adapters are efficient and swappable tools allowing model adaption to specific tasks, domains, and languages. Each language adapter is trained on a monolingual corpus on its target language using a language modelling objective, and these adapters can be used to extend original language model parameters with additional parameters

[1] https://www.taus.net

specific to the target language. This has been shown to lead to an increase in model performance on a series of languages.

Adapters are trained individually with the original model parameters frozen around them. This makes various adapters compatible with each other, allowing combinations such as adapter stacks. Given a task and a target language a, my goal is to stack language adapter LA_b for language b on top of language adapter LA_a for language a to increase task performance on language a.[2]

In this work, I set out to address the following research questions.

- Can the stacking of language adapters contribute to increased model performance on downstream tasks?
- Which linguistic factors determine which language adapters are useful to combine?
    1. Is typological similarity between the languages a good predictor for which language adapters combine well?
    2. Are simple corpus-based measures such as type-token ratio good estimators for morphological complexity of particular languages?
    3. Does similarity in terms of morphological complexity well represent typological similarity between languages?
- Which extra-linguistic factors determine which language adapters are useful to combine?
    1. Do training-related factors, such as the size of the training data for the two language adapters determine the contribution of language adapter stacking to the success on downstream tasks?
    2. Do other factors related to the training data, such as shared scripts, determine the contribution of language adapter stacking to the success on downstream tasks?
- Which languages benefit the most from stacking language adapters?
    1. Does model performance increase on all languages or only on low resource languages?
    2. Does model performance increase on zero-shot languages, i.e., languages that the language model has not seen during its pretraining?

At the end of Chapter 2, once the theoretical background is provided for this thesis, I also outline the hypothetical answers to these questions.

1.2 Structure

In the following, I list the individual chapters of this thesis, describing their respective contents.

In Chapter 2, I introduce theoretical foundations behind important concepts for my work, such as language modelling and pretrained transformer-based multilingual

[2]While it is theoretically possible to stack adapters of language a on top of language b, due to the large number of experiments, I decided to limit investigation to this setup. Also because based on the results shown in Table B.1 in Appendix B, the order does not seem to change the trends significantly.

language models, cross-lingual transfer, and adapter modules as a technique of model adaptation. I also outline related work on the effect of various linguistic and extra-linguistic factors on language modelling with special attention to typological features.

In Chapter 3, I describe the methods used to address my research questions. I present multilingual BERT, the multilingual language model I use in my experiments, and I also show how language adapters can be added between its layers. Then, I define corpus-driven measures of morphological complexity, the typological feature that I use to create adapter stacks in the experiments, and show how it can be estimated using a parallel corpus. Finally, I present the evaluation task, cross-lingual sentence retrieval, and relevant datasets.

In Chapter 4, I describe how I operationalise my research questions and how I prepared for the experiments. I also describe how I arrived at the sample of languages I use for the experiments. As I conducted a large number of experiments, I then illustrate the individual setups on a single language in order to make further discussion more straightforward.

In Chapter 5, I present the results of the experiments for each language in my sample. Due to the large number of experiments, I first focus on showing trends such as experiments with the best and worst scores. Afterwards, I focus on low resource languages, the ones that I hypothesise potential improvements can be reached using stacked language adapters.

In Chapter 6, I discuss the results in more detail. In the first half of the chapter, I address my research questions. In the remaining parts, I carry out both a qualitative and quantitative error analysis.

Finally, in Chapter 7, I briefly conclude the project while highlighting my main findings.

Chapter 2

Background

In my thesis, I set out to induce cross-lingual transfer across typologically similar languages in a multilingual language model using stacked adapters. In this chapter, I introduce the theoretical foundations that underlie this work, as well as prior research that provides context to my thesis.

First, I present principles of language modelling with focus on multilingual context. In the section afterwards, I describe adapters, efficient model adaptation modules that form the basis of my experiments. In Section 2.3, I outline the importance of linguistic and extra-linguistic factors for language modelling with special focus on factors related to typology. Moreover, I introduce prior research investigating the role of inflectional morphology with regards to language modelling.

Finally, I address the research questions posed in Chapter 1, providing clear hypotheses based on theoretical foundations that are provided in this chapter.

2.1 Multilingual language models

In this section, I describe language modelling in NLP with special attention to pretrained transformer-based massively multilingual language models.

In Section 2.1.1, I introduce the basic principles of language modelling. In Section 2.1.2, I show how language models are used in cross-lingual scenarios. Finally, in Section 2.1.3, I provide an overview of the training and the basic components of pretrained transformer-based language models.

2.1.1 Language modelling

Language models assign probability distribution to a sequence of words. During their training, they learn to allocate high probabilities to sequences that are plausible (grammatical), and low probabilities to those that are not. The role of context is essential in natural languages even across long distances. The better the model can capture dependencies between elements in a sequence, even when they are far away from each other, the better it is expected to perform.

Language models learn to assign *surprisal*, i.e., negative log-likelihood to sequences. Surprisal reflects how difficult a particular segment is to model for the language model. Longer and more complicated segments inherently have higher surprisal values, but surprisal also correlates with how well a language model might capture characteristics of the language.

Surprisal, or negative log-likelihood NLL is calculated based on the equation in 2.1, adopted from Mielke et al. (2019)

$$NLL(s) = -log_2 P(s), \tag{2.1}$$

for sequence s where P stands for the probability of the sequence.

The simplest language models, so-called *n-gram* models, are trained on predicting the next word based on the preceding $n-1$ words. N-gram models learn the relative frequency of collocations in the training corpus, learning which sequences are more frequent than others. Bigram (2-models) language models predict the current word based on the previous word, trigram (3-gram) models predict based on the previous two words, and so on. While predicting only based on the previous $n-1$ words is a simplistic way to model natural language, n-gram models can still be very useful in disambiguation tasks.

Ambiguity is rampant in language, but this most often does not pose a difficulty for humans. Context in most cases can help ascertain the correct interpretation. N-grams are capable of grasping a certain amount of contextual information that can help them to arrive to a correct reading of sentences such as 1.

(1) A boltnál vár rád. (Hungarian)
the shop.AT wait.3.SG.PRES you.ON
He/she is waiting for you at the shop.

Homographs like Hungarian 'vár' *wait.3.SG.PRES* or *castle.SG*, can often be distinguished between only using the local context. In this case, it is clear that the correct interpretation is the one involving the verb. N-grams are capable of capturing this local context, but they struggle with further links.

Language modelling capabilities improved significantly with the advent of neural language models due to their superior ability to capture long-distance dependencies. Besides a capability to assign probabilities, these language models also allow the creation of contextualised embeddings. These embeddings reflect not only the abstract lexical meaning of a token, but also its semantics within the context. These contextual embeddings can be used as inputs to many NLP tasks.

State-of-the-art language models for years have been based on the transformer architecture proposed by Vaswani et al. (2017). These models, such as the multilingual BERT model I use for my experiments employ the attention mechanism to be able to link words across arbitrarily long distances in a given segment.

2.1.2 Cross-lingual language modelling

Pretrained massively multilingual models represent a variant of language models that are jointly trained on corpora from 100+ languages. These models learn to construct a shared embedding space for all the languages involved even without explicit cross-lingual signals (Wu and Dredze, 2019; Jones et al., 2021). This shared embedding space allows a transfer of model knowledge across languages, enabling zero-shot cross-lingual transfer. It also makes it possible for the model to solve cross-lingual tasks.

Zero-shot cross-lingual transfer

Pretrained language models acquire general-purpose linguistic knowledge during their pretraining that can be leveraged to adapt (*fine-tune*) the models to specific scenarios. These might include a specific domain, task, or language. This is the so-called *pretrain and fine-tune* paradigm that is so prevalent in current NLP.

Fine-tuning typically requires scenario-specific training data. However, for most tasks, especially semantically complex ones such as question-answering, there is often no annotated data available for low resource languages. This is where multilingual models have an advantage over monolingual models. They can be fine-tuned on training data in a single source language, and, through their shared multilingual embedding space, they can transfer the task-specific knowledge to the target task in any of the other languages the model is familiar with. This is called *zero-shot cross-lingual transfer*, and it is an essential feature of multilingual models (de Vries et al., 2022; Lin et al., 2019).

The capabilities of various multilingual models to carry out zero-shot transfer have been assessed on a variety of tasks. Among all, these include part-of-speech tagging, morphological tagging, named entity recognition, or dependency parsing, that do not require in-depth semantic knowledge (Pires et al., 2019; Wu and Dredze, 2019; de Vries et al., 2022; Lin et al., 2019; Pfeiffer et al., 2020b). Tasks requiring semantic knowledge have also been investigated, including entity linking, question-answering, document classification, paraphrase identification, commonsense reasoning and natural language inference (Wu and Dredze, 2019; Pfeiffer et al., 2020b; Ponti et al., 2020; Artetxe et al., 2020; Lin et al., 2019).

Multilingual benchmarks have also been proposed that measure how well multilingual models can carry out zero-shot transfer. These include an assortment of tasks with various difficulties, evaluating zero-shot transfer from English to other target languages. Notable examples are XGLUE (Liang et al., 2020), XTREME (Hu et al., 2020), and XTREME-R (Ruder et al., 2021) (see Section 3.3 for a discussion on the latter two benchmarks).

Cross-lingual tasks

The capability of multilingual language models to map input from different languages to the same shared embedding space enables them to be used for a variety of cross-lingual applications.

One such task is *parallel text mining*, where embeddings can be used to identify parallel sentence pairs in parallel monolingual corpora. This is especially useful for generating data to train or improve neural machine translation models. Neural machine translators are trained on parallel corpora, and the better segments align in these corpora, the higher quality translations it can create.

This is a use case that is especially relevant for TAUS, the company where I pursue my thesis internship. In the past, TAUS has been trading with parallel corpora, helping other companies improve their machine translation solutions. Nowadays, TAUS focuses on a new product called Data-Enhanced Machine Translation (DeMT™), where they carry out the adaptation of the machine translation solution themselves.

BUCC (*Building and Using Comparable Corpora*) is a shared task that can be used to measure the potential of multilingual models in parallel text mining (Zweigenbaum et al., 2018). Competing models have to identify correct sentence pairs across monolingual texts. BUCC is also incorporated as a standalone task in the XTREME benchmark

mentioned in the previous section (Hu et al., 2020). Its main drawback is that it only contains data for five high resource languages with little typological diversity: English, German, French, Russian and Chinese. This means the task cannot be used to assess the capabilities of multilingual models across typologically diverse and lower resource languages. Existing multilingual models tend to perform well on BUCC (Ruder et al., 2021).

Another task is the *cross-lingual sentence retrieval* – also called *translation pair detection* – task, which is in many ways similar to parallel text mining. In this scenario, the model has to be able to create well-aligning representations across languages that allow it to link translations to target sentences. As it can be contextualised as building a parallel corpus, cross-lingual sentence retrieval is also a task that is relevant to TAUS. It also assesses the capabilities of the language model to create well-aligned representations across languages.

The most well-known test set for cross-lingual sentence retrieval is the Tatoeba task dataset (Artetxe and Schwenk, 2019). Tatoeba is an open-source collection of English sentences and its translations that are provided by volunteers.[1] The Tatoeba task involves generating sentence-level representations using a specific language model, and returning translation pairs based on the cosine similarity of their representations. Subsets of the Tatoeba task dataset are integrated in both the XTREME and XTREME-R benchmarks. The drawback of the task is that the Tatoeba task sets are not completely parallel across all languages, making direct comparison of results difficult (Hu et al., 2020; Ruder et al., 2021).

2.1.3 Pretrained transformer-based language models

Transformer-based language models create contextual embeddings for elements of an input sequence, most commonly for tokens of a sentence. In this section, I present an overview of the training objectives used to pretrain these transformer-based models. I also describe components of these models, and how they are used in processing their input data.

Pretraining objectives

Multilingual models are capable of generalising across languages (Pires et al., 2019; Wu and Dredze, 2019). This is a curious phenomenon given that the such models typically share training objectives with monolingual language models that do not explicitly facilitate these types of links.

The simplest language models, n-gram models, are trained on predicting what the most likely word is given the previous $n-1$ words (see Section 2.1.1). However, transformer-based language models are more sophisticated, allowing the potential to capture long-distance dependencies within a natural language sequence from either directions. To adapt to this property, the training objective needs to be altered.

Linear next word prediction is replaced by a training objective called *masked language modelling* (Devlin et al., 2019; Liu et al., 2019; Conneau et al., 2020). Masked language modelling replicates the cloze test used in second language teaching and pyscholinguistic experiments. The test involves filling in a gap in a sentence based on the content of the sentence. In this, the broader context provided in the sentence can normally be used to make correct predictions.

[1] https://tatoeba.org/en/

As an illustration, take the example in sentence (2).

(2) Today, I went to the _____ and I bought bread and sugar.

Just like language learners or experimental subjects can guess words such as *store*, *shop*, or *supermarket* from the context, transformer-based language models also learn to use linguistic, semantic and contextual cues to make the correct predictions.

In the pretraining of BERT, one of the first transformer-based models, 15% of all tokens are randomly selected in the training data. 80% of the selected tokens are masked with a placeholder token, 10% of them are replaced with a random vocabulary item, and in the final 10% of the cases, the token is unchanged. The training objective of masked language modelling is to predict the original input tokens in the place of the selected tokens (Devlin et al., 2019). To allow successful pretraining, massive amounts of training corpora are required. The multilingual BERT model I use in my experiments, for instance, is training on the articles of the 104 largest languages on Wikipedia (Wu and Dredze, 2020).

Besides masked language modelling, certain models are also trained on additional training objectives. These aim to capture features of language that are not captured by masked language modelling. BERT and multilingual BERT both use *next sentence prediction*, a binary classification task of deciding whether two input sentences follow each other or not. This training objective was selected to help the language model understand the relationship between sentences, which may come useful in question-answering and natural language inference tasks (Devlin et al., 2019). In practice, however, many later models only use the masked language modelling training objective, debating the contribution of next sentence prediction (Liu et al., 2019).

Path of the input data inside the model

Input segments (typically a single sentence or a pair of sentences) are first broken down into a sequence of subword tokens whose representations the model acquires during pretraining. By dividing input words into smaller subword tokens, language models are capable of representing words that do not occur in their training data. These subword tokens are generated by algorithms such as the WordPiece algorithm used in the multilingual BERT model (Schuster and Nakajima, 2012).

WordPiece works like similar algorithms, such as Byte-Pair Encoding. It requires a training corpus that it splits up into individual words, typically based on whitespaces. Once this is done, a set of unique words is created with their frequencies established. Then, WordPiece initialises a base vocabulary using all the symbols that appear in the training corpus. These symbols serve as the character set that the language model can roll back to in order to represent novel words.

Then, the algorithm starts merging symbols according to the principle of maximising the likelihood of the training data. WordPiece calculates the probability of a symbol pair s_1, s_2 divided by the probability of the individual symbols s_1 and s_2 (see 2.2).

$$\frac{P(s_1, s_2)}{P(s_1)P(s_2)} \tag{2.2}$$

The symbol pair with the highest probability is merged and added to the vocabulary, and the merging is continued until the algorithm terminates. WordPiece terminates

either when a certain number of merges has been performed, or when a certain vocabulary size is reached. These counts are hyperparameters set prior to the use of the algorithm.

Special tokens, such as a prepending CLS token are added to each input sentence. When sentence-level representations are required, common practice is to either use the CLS token (Devlin et al., 2019) or to pool the subword token embeddings in the sentence, for instance by averaging them (Hu et al., 2020).

In the embedding layer of the model, each special token and subword token is replaced by the corresponding static embedding that the language model learnt during its pretraining. These static subword embeddings are then propagated further in the higher layers of the language model. Each layer is made up of transformer blocks which consist of multiple self-attention heads. These regulate the inclusion of contextual information from the surrounding subword tokens into the embeddings of each individual subword token.

Layer by layer, representations of subword tokens may come to incorporate more and more relevant features about the subword tokens in their context. After passing through the final model layer, subword representations can be used in downstream tasks. Often this is done by adding a classifier on top of the language model.

Since the training objectives used in pretraining serve only to endow the model with general-purpose linguistic knowledge, most applications involve adapting the model to target tasks using additional task-specific training data. In the next section, I discuss these fine-tuning techniques.

2.2 Adapter modules

Pretrained language models as described in Section 2.1 are powerful learners acquiring sophisticated linguistic knowledge during their pretraining. This knowledge, however, is rather generalized and often not too useful when it comes to solving downstream tasks. There are various model adaptation techniques that address this issue, using task-specific data to tune pretrained language models to a target scenario. The most well-known model adaptation technique is *full fine-tuning*, which involves modifying model weights to a specific task.

In this section, I introduce the mechanics and the downsides of full fine-tuning. I also describe lightweight adapter modules, an alternative technique that was proposed to address many of these downsides.

2.2.1 Fully fine-tuning

As mentioned in section 2.1.2, the most common framework in using current PLMs is the so-called *pretrain and fine-tune* paradigm. Under this paradigm, language models are pretrained with a language modelling objective, acquiring robust general-purpose knowledge. Fine-tuning leverages this knowledge. Taking a task-specific dataset, fine-tuning involves inducing gradients from downstream training examples, updating all parameters of the model (Liu et al., 2021b).

Fully fine-tuning is a simple and effective approach, but it comes with certain downsides. It has a tendency to lead to overfitting, and when the training dataset is too small, the language model might not be able to learn in a stable manner.

Fine-tuning might also result in the language model losing knowledge it acquired during its pretraining or in a previous fine-tuning in a process called *catastrophic forgetting*. The main cause of catastrophic forgetting is that fine-tuning affects all model parameters. It has the potential of obscuring the original information that was encoded in these weights (Liu et al., 2021b; Pfeiffer et al., 2021a).

Catastrophic forgetting might also prevent the success of multi-task learning. In principle, by jointly training on training data from multiple tasks, multi-task learning enables a model to generalize between related tasks, for instance between natural language inference and commonsense reasoning. However, joint training of the same parameters might lead to the model overfitting on low resource tasks and underfitting on high resource ones (Pfeiffer et al., 2021a).

2.2.2 Adapters

Adapters are efficient and modular fine-tuning methods proposed as an alternative to fully fine-tuning. They are injected between language models layers with the original model weights frozen. During model adaptation, only the parameters encapsulated inside the adapters are modified. Since adapters trained with the same language model create compatible representations, they are easy to swap and combine. This property makes them useful in industry and research as well (Houlsby et al., 2019).

There is evidence that adapters are more parameter efficient, less sensitive to hyperparameter choice during training, less prone to overfitting, and that they can outperform fully fine-tuning (Bapna and Firat, 2019; He et al., 2021a; Pfeiffer et al., 2020b, 2021a). Adapters have further advantage in that they provide a scalable solution due to their relatively small size, corresponding to around 1% of model weights in a pretrained language model. They preserve original model weights, so they facilitate the distribution of adapted pretrained language models as only the adapter weights need to be exchanged. Since adapters are trained separately, and can be freely combined later, model adaptation via adapters is resistant to catastrophic forgetting, one of the major issues with full fine-tuning (Pfeiffer et al., 2020a).

In this section, I introduce how adapters are commonly trained and used, and give a formal description of them. Finally, I describe how adapters can be combined.

Adapter use

Adapters are standalone modules that can be added to pretrained language models as an alternative to fully fine-tuning them. They have been used in adapting pretrained language models to specific tasks, domains, and languages. Adapters learn compatible representations with the original model and each other thanks to the fact that the original model weights are not frozen during their training. These representations make it possible for adapters to be freely added and removed from the language model, or to be replaced and combined with each other (Pfeiffer et al., 2020a).

The process of creating new adapters is similar across task, language, and domain adapters. Prior to adapter training, the original language model weights are frozen and adapter weights are inserted between the language model layers. Then, a so-called model head, i.e., a classifier is placed on top of the language model relevant to the training objective: a downstream task for a task adapter, and masked language modelling for a language or domain adapter (see Section 2.1.1 for details). After this, adapter weights change until a certain training condition is reached.

Adapters formally

Adapters aim to extend existing language model capacity, preserving the original model weights. If a language model is formally described as a function weights $\mathbf{w}$: $\phi_{\mathbf{w}}(\mathbf{x})$, then adapters are a small set of additional weights $\mathbf{v}$, $|\mathbf{v}| \ll |\mathbf{w}|$. The adapted language model can be described as $\psi_{\mathbf{w},\mathbf{v}}(\mathbf{x})$, with original weights $\mathbf{w}$ carried over from the original PLM. Initial parameters prior to adapter training $\mathbf{v_0}$ are determined in a way that $\psi_{\mathbf{w},\mathbf{v_0}}(\mathbf{x}) \approx \phi_{\mathbf{w}}(\mathbf{x})$. During training, the original model weights $\mathbf{w}$ are frozen, and only the adapter weights $\mathbf{v}$ are adjusted. Since the original parameters are unaffected by adapter training, adapters allow model adaptation in a modular way. All one needs to do is replace the adapter modules (and the set of weights $\mathbf{v}$) with new adapter modules and new $\mathbf{v}'$ weights, $|\mathbf{v}'| \approx |\mathbf{v}|$.

Adapters in NLP were first proposed as bottleneck modules inserted between layers of pretrained transformer-based language models. Various implementations of adapters come with certain differences. However, many approaches follow a bottleneck architecture. These adapters take output from the previous transformer block, carry out a down-projection, apply a non-linear function, and up-project the resulting representation, matching the output with the dimensionality of the original language model.[2] Adapters typically also maintain a residual connection that allows the language model signal to pass through the adapter unaffected by it.

Equation 2.3 defines the structure of adapter A at language model layer l according to Pfeiffer et al. (2020b).

$$\mathsf{A}_l(\mathbf{h}_l, \mathbf{r}_l) = \mathbf{U}_l(\mathsf{ReLU}(\mathbf{D}_l(\mathbf{h}_l))) + \mathbf{r}_l \tag{2.3}$$

The down-projection is $\mathbf{D} \in \mathbb{R}^{h \times d}$. h is the dimensionality of the language model and d is the dimensionality of the adapter module, typically half of the dimensionality of the language model. The non-linearity in this example is ReLU, while the up-projection is $\mathbf{U} \in \mathbb{R}^{d \times h}$. Finally, in every adapter layer there is also a residual connection $\mathbf{r}$ (Pfeiffer et al., 2020b).

Adapter combinations

Since adapter modules are encapsulated between frozen language model layers during their training, their output is compatible with both the original model weights and each other. Pfeiffer et al. (2020b) utilizes this compatibility in their MAD-X approach.

MAD-X facilitates zero-shot cross-lingual transfer using adapters. Take a specific task t with training data available in source language a, and a target language b. Prior to training the task adapters TA_t, source language adapters LA_a are added to each language model layer. The untrained task adapters TA_t are each stacked on top of source language adapters LA_a. This stacking means that apart from the residual connection, the input signal to each task adapter TA_t comes from a source language adapter LA_a. At inference time, zero-shot transfer can take place by replacing each source language adapter LA_a with target language adapters LA_b. Thanks to the compatibility of adapter outputs, task adapters TA_t can also take their inputs from target language adapters LA_b that they were not combined with during training.

[2] While it is beyond the scope of this thesis, alternative adapter architectures were also proposed that do not use the same bottleneck setup described here. These include Prefix Tuning (Li and Liang, 2021), Mix-and-Match adapters (He et al., 2021b), and Compacters (Mahabadi et al., 2021).

Stacking is only one of several ways adapters can be combined. It is, however, a simple and straightforward approach. It is already shown by the MAD-X framework of Pfeiffer et al. (2020b) to allow representations from multiple adapters to combine to improve performance on a target scenario. Given that structural and lexical similarity between languages may encourage the transfer of lexical information (see Section 2.1.2 and 2.3.3), stacking language adapters of a well-chosen pair of high and low resource language might allow a pretrained language model to apply knowledge from the high resource language to the benefit of the low resource one. This might result in comparatively higher performance on low resource languages. This is the main foundation that prompted my research questions as well.

In the next section, I describe what types of similarity between languages might be conducive for transferring linguistic information.

2.3 Typology in language modelling

Despite joint pretraining on corpora from multiple languages at the same time, pretrained transformer-based multilingual language models do not perform equally well for all languages. Language modelling performance correlates with the amount of training data that is available for specific languages. High resource languages, such as English, Spanish, German, or Chinese, tend to have plenty of texts that can be used to pretrain language models. These languages have some of the largest Wikipedia sizes (Wu and Dredze, 2020) and the largest data sizes in CommonCrawl (Conneau et al., 2020) as well. Most of the world's languages lack behind in the size of available data, impacting the level of linguistic knowledge that can be captured for them by language models.

However, language modelling performance does not solely depend on training data size. Language-specific properties, such as orthography, lexical overlap with other languages, or typological features may also affect the capabilities of language models.

Typology also affects the efficacy of cross-lingual transfer between languages. There is evidence from prior research that typological similarity between languages plays a role in cross-lingual transfer, the phenomenon of generalizing linguistic information from one language to another inside multilingual models (see Section 2.1.2 for more details).

In the following, I describe factors that correlate with language modelling performance with special focus on typological features. Following this, I show how these features can be quantified. I continue by looking at how cross-lingual transfer is impacted by typological similarity. Finally, I introduce three papers that investigate the role of inflectional morphology on language modelling performance (Cotterell et al., 2018; Mielke et al., 2019; Park et al., 2021).

2.3.1 Factors impacting language modelling difficulty

Massively multilingual language models such as multilingual BERT (Devlin et al., 2019) and XLM-RoBERTa (XLM-R) (Conneau et al., 2020) can cover 100+ different languages, using them also saves effort in creating monolingual models and enables cross-lingual tasks. However, there are several factors that limit their performance on different languages. In this section, I provide a few of these.

Training data size

Not all languages have the same training data sizes available. multilingual BERT, for instance, is pretrained on Wikipedia for 104 languages. The size of Wikipedias for different languages, however, is widely divergent. While English Wikipedia has tens of gigabytes of texts, the data size for other high resource languages is around 3-6 gigabytes. For medium resource languages, such as Finnish and Turkish, the Wikipedia size does not even reach 1 gigabyte, and for a truly low resource language, for example, Yoruba, it is less than 0.01 gigabytes (Wu and Dredze, 2020).

XLM-RoBERTa, another transformer-based multilingual language model extends the training data of multilingual BERT by using an additional 2.5 terabytes of texts from CommonCrawl. This results in training data sizes ranging from 300 gigabytes for English and Russian, 20 gigabytes for Turkish, and 0.1 gigabytes for the smallest languages, such as Xhosa, Assamese, and Oromo (Conneau et al., 2020). While there are attempts to balance the training data of different languages by under- and over-sampling them, performance on under-represented languages typically suffer (Wu and Dredze, 2020).

Tokenizer coverage

As mentioned in Section 2.1.3, pretrained language models use a data-driven subword tokenization approach to segment input data. Data-driven subword tokenization addresses problems that arise when using alternative approaches to input segmentation.

Segmenting input text on word boundaries (for instance, on whitespaces) would result in the model being incapable of representing words that do not appear in the training data, thus making it less useful when applied to novel data. Character-based tokenization, on the other hand, would result in too long sequences to handle. Additionally, it is challenging to learn meaningful representations for characters, since in themselves they do not carry meaning.

Finally, tokenization based on morphological boundaries could result in creating superior contextual embeddings, but morphological parsers can also struggle with novel words and ambiguous analyses. They also add an overhead to processing new input sequences that data-driven parsers simply do not.

The downside of data-driven subword tokenization that the algorithms result in tokens that are not linguistically meaningful. Learning good representations for these can be a challenging task. Consider the examples in (3) and (4).[3]

(3) a. screen *(screen)*
b. sun ##sc ##reen *(sunscreen)*

(4) a. glass ##es *(glasses)*
b. sung ##lasse ##s *(sunglasses)*

We can discover considerable differences between how the subword tokenizer of multilingual BERT breaks down individual words, even when they share many of the same elements. Contrasting 3a and 4b, for instance, we can see that the word *screen* can either be a separate subword screen, corresponding to an actual English word, or that it can be broken down into two subword tokens made up of random n-grams, ##sc ##reen.

[3] All examples were generated using the multilingual BERT tokenizer, following the convention of using double # signs to represent subword tokens that are not word initial.

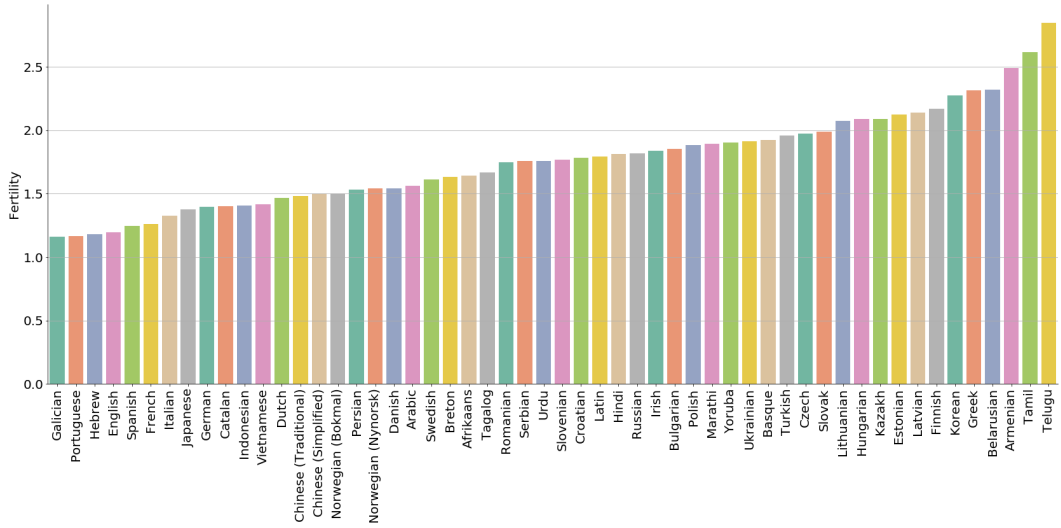


Figure 2.1: Average number of subwords generated per tokenized word (*fertility*) by the WordPiece tokenizer of multilingual BERT for various languages. On average, the higher this fertility value is, the more segmented words of a language are. Figure taken from Ács (2019).

The word *glasses* in 4a is split into subword tokens that correspond to actual English morphemes: glass ##es. However, when *glasses* forms the second half a compound noun, *sunglasses*, it is split in such as way that not even the word boundaries are preserved within the compound: sung | ##lasse ##s. Finally, when comparing how the compound element *sun* is processed by the subword tokenizer, we can discover that depending on the context, it might be preserved (in 3b), or it might be take on a form that corresponds to a completely different word, *sung* (in 4b).

Examples in 3 and 4 show how a data-driven subword tokenizer might obscure morpheme identity even for English, a high resource language with a relatively simple morphology. The fact that the same morphemes are split in a variety of different ways depending on the larger word they appear in makes distributional patterns that language models learn from all the more difficult to acquire.

The original BERT model has a subword vocabulary of 30,000 tokens which form a superset of all characters in the Latin alphabet (Devlin et al., 2019). Multilingual language models such as multilingual BERT have a shared subword vocabulary that is meant to represent all languages the model has encountered during training. multilingual BERT, for instance, has a vocabulary size of 115,000 subword tokens. The vocabulary of both these models is created using WordPiece, one of the many data-driven subword tokenization algorithms (Schuster and Nakajima, 2012).

Since data-driven subword tokenization uses statistical information to create a vocabulary, languages with larger training corpora will have a superior tokenizer coverage. This manifests in both in terms of number of subwords and the relative frequency of subwords. Wu and Dredze (2020), for instance, show that for languages with smaller training data sizes, the multilingual BERT vocabulary contains much lower frequency items than for high resource languages. This implies that the WordPiece tokenizer of multilingual BERT maps worse to low resource languages.

There are other measures that can quantify the coverage the model tokenizer has on

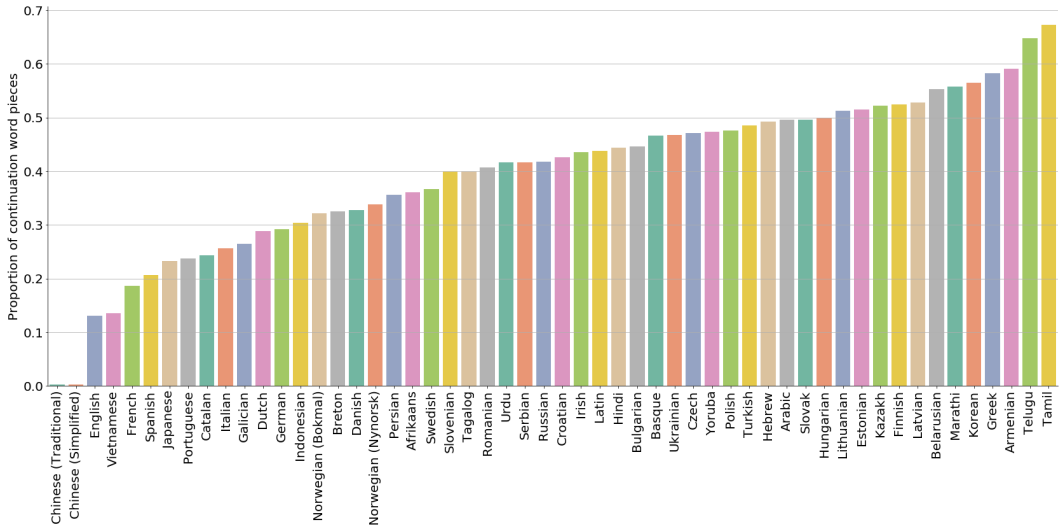


Figure 2.2: Proportion of non-initial (continuation) subword tokens per language in the multilingual BERT tokenizer. As Chinese characters are segmented before the model tokenization, the proportion of continuation subword tokens is particularly low, 0.2%. Figure taken from Ács (2019).

a particular language. One of these measures is subword *fertility*. This score measures the average number of subwords tokens are split into (Ács, 2019; Rust et al., 2021). Lower fertility values indicate a closer to 1–to–1 relation between subword tokens and vocabulary items of a specific language. High fertility values might indicate that words of a particular language are over-segmented and make it likely that instead of appropriate tokenizer coverage, most subwords relevant to the language are missing from the shared model vocabulary (see Figure 2.1).

Another potential measure of tokenizer coverage is the proportion of subword tokens that are not word-initial (i.e., starting with ##) (Ács, 2019) (see Figure 2.2). A related metric is the proportion of words that are continued over at least two subword tokens that are not word-initial (Rust et al., 2021). Both these measures, just like fertility, reflect a certain aggressiveness in how a data-driven subword tokenizer algorithm segments input from different languages. In practice, high fertility values often indicate that the model vocabulary does not adequately match the vocabulary of the language, thus the tokenizer has to roll back to segmentation based on characters or subwords that do not match up with meaningful morphemes in the target language.

However, having high fertility or a proportion of subword tokens that are not word-initial does not necessarily reflect how well or badly a tokenizer fits a particular language. Language models that are trained well on a morphologically complex language, for instance, should be capable of splitting morphologically complex words into individual morphemes to better capture their meaning. For languages with more complex inflectional morphology, this could conceivable result in comparatively high values for all three measures described above. It is telling that in both Figure 2.1 and 2.2, the highest fertility values and the highest proportion of subword tokens that are not word-initial are both associated with morphologically complex, agglutinative languages, such as Tamil and Telugu.

Another measure of tokenizer coverage is the amount of lexical overlap between the shared subword vocabulary of multilingual BERT and of the vocabulary of different languages. Pfeiffer et al. (2021b) show that this measure seems to correlate with zero-shot cross-lingual transfer capabilities of multilingual BERT. For most languages seen during training, the proportion of lexically overlapping tokens is around 25% or more (Wu and Dredze, 2020). In the case of languages that are not written with the Latin alphabet, this proportion is generally lower, with a minimum of 6% of Arabic (Pfeiffer et al., 2021b). In the case of languages not seen during training, many languages written with the Latin script reach a coverage close to 50% with languages such as Amharic with its own Ge'ez script reaching only 13% (Pfeiffer et al., 2021b).

Tokenizer coverage can also be measured by the ratio of tokens that contain characters the multilingual BERT tokenizer cannot represent and has to be replaced by an UNK (unknown) token. This is generally a very low percentage for languages written with scripts seen during training: 0% for English (written with the Latin script), 1% for Arabic (written with the Devanagari script), 2% for Hindi (written with the Devanagari script) (Pfeiffer et al., 2021b).

Tokenizer coverage seem to generally depend on the script a language uses. The most common script in the training data of multilingual BERT is the Latin alphabet, covering 78.21% of all subword tokens. This is followed by Chinese, Japanese and Korean[4] with 12.49%, Cyrillic with 11.38%, and various Indian alphabets with 5.47% (Ács, 2019). Languages written with scripts that are highly represented in the model vocabulary are likely to have a higher tokenizer coverage compared to ones where the relative rarity of the script results in less subwords.

Typological features

Linguistic typology is the systematic investigation of variation that is present in the world's languages. In NLP, there is growing attention on the impact structural differences between languages have on the difficulty of language modelling.

Typological features that are investigated typically belong to the domain of morphosyntax. During their pretraining, multilingual models learn from distributional patterns. Languages with less rigid syntactic rules have less predictable token sequences, which might result in a difficulty capturing these distributional patterns. Morphologically rich languages, on the other hand, might pose a challenge due to the internal complexity of their word structures. As multilingual language models have a shared subword vocabulary between all languages, individual morphemes of specific languages might not correspond clearly to subword tokens. This is especially problematic for low resource languages whose words are likely poorly covered by subword tokenizers anyway. Lack of consistency on how individual word forms are broken down into subwords makes capturing distributional patterns even more difficult.

2.3.2 Quantifying typological features

To measure how typological features might impact language modelling difficulty, we need to quantify these features. Typological features might be measured using expert

[4]Macro-category by Ács (2019).

judgement, NLP tools, such as parsers, or by using corpus-driven measures that work solely on the basis of the statistical characteristics of the texts in the language.

The most precise way to capture typological features for languages is to use expert judgement. In NLP, this information often comes from aggregate resources such as the World Atlas of Linguistic Structures (WALS) (Dryer and Haspelmath, 2013).[5] While these judgements are typically accurate and reliable, their coverage is far from full. Bentz et al. (2016), for instance, isolate 28 features from WALS pertaining to morphology. Out of 1713 languages with at least 1 feature value identified, 34 have values for at least 27 of these features, and only 10 have all the feature values identified. In practice, it is more common to select one or two features from these expert resources that are annotated for a larger number of languages. Such features might describe whether the inflectional morphology is overwhelmingly prefixing or suffixing, or whether the language is polysynthetic or not. However, these distinctions are typically not too fine-grained.

Expert judgement also informs Universal Dependencies (Nivre et al., 2020), an framework for annotating part of speech tags, morphological features, and syntactic dependencies. Universal Dependencies is an important resource for quantifying typological features in the domain of morphosyntax. Yet, it is not without its drawbacks. It only covers 100 languages, and there is considerable inconsistency with which morphological features are applied to different languages. Universal Dependencies can be valuable though when it comes to quantifying structural similarity between languages by counting the dependency links that exist in their treebanks (Bjerva et al., 2019).

Expert annotated dependency links are only available for a subset of languages. A less immediate use of expert judgement is to use NLP tools to measure a certain typological property. Parsers such as UDPipe (Straka et al., 2016), spaCy and Stanza are trained on annotated data and are capable of generalizing to new texts. However, this annotated data is not available for many languages.

Corpus-based measures are derived from the distribution of words inside a given text. As long as there is monolingual data available for a language, these measures can be quantified. Thus they can be widely applied. However, the use of corpus-based measures requires caution. It is crucial to be apply them on parallel texts of the same domain, since the values of these measures are influenced by stylistic factors, such as the size of the vocabulary and the length of words used in a given text.

Bentz et al. (2016) introduces and evaluates a collection of such corpus-based measures. The authors compare how well morphological complexity measures based on expert judgement derived from WALS correlate with a number of corpus-driven measures.

Bentz and colleagues demonstrate that all the complexity measures they consider significantly correlate with each other. They also discuss advantages and disadvantages of the various measures. Complexity measures based on WALS might be accurate, but the coverage is sparse and coarse-grained. Corpus-based measures may give much more fine-grained results that are cross-linguistically comparable, but they may not be able to distinguish between patterns of regular processes of word formation (more indicative of classically agglutinative languages) and irregular ones (more characteristic of fusional languages).

To get the value of the expert measure, the authors first identify 28 features from WALS for describing morphology. They order feature values from quantifying less mor-

[5] https://wals.info

phologically complex to more, normalize their values, and average to get a single number. The higher this number is for a language, the higher its morphological complexity is. The method of Bentz and colleagues is supported by the fact that languages with rich morphology, such as Turkish, Basque and Hungarian have considerably higher scores than languages with more isolating tendencies, such as English, Maori, and Vietnamese (Bentz et al., 2016).

Among corpus-based measures, the conceptually simplest one is type-token ratio. For its definition, Bentz et al. (2016) define a word as a string of alphanumeric characters delimited by whitespace characters. Given a text, type-token ratio is calculated by adding up the number of distinct word tokens, and dividing this number by the sum of all word tokens. Rich inflectional morphology results in the proliferation of distinct word types. Thus if type-token ratios are determined based on the same text for multiple languages, more morphologically complex languages will have a higher value. However, the number of distinct word types, i.e., the vocabulary of a text might differ based on style and register as well.

Another measure Bentz et al. (2016) introduce build on the information-theoretical concept of entropy. Languages with a rich morphology are more likely to have a higher information content encapsulated in their words than ones with simpler morphology. Given a vocabulary of word types $\mathcal{V} = w_1, w_2, \ldots, w_V$ of size $\mathcal{V} = |V|$, the entropy H inside a text T is estimated based on the formula in 2.4 (taken from (Bentz et al., 2016)):

$$H(T) = -\sum_{i=1}^{V} p(w_i) log_2(p(w_i)) \tag{2.4}$$

While Bentz et al. (2016) use a more sophisticated estimator of word type probabilities $p(w_i)$, they propose that the simplest way to $p(w_i)$ is to use a *maximum likelihood estimator*. This maximum likelihood calculates $p(w_i)$ by normalizing type frequencies by the overall token count.

2.3.3 Language similarity in cross-lingual transfer

As introduced in Section 2.1.2, cross-lingual transfer is a subset of transfer learning, and it refers to the phenomenon of carrying over linguistic information from one language to another. It is enabled by the cross-lingual representation space that emerges inside multilingual language models during their joint pretraining on data from multiple languages (Section 2.1.2). The success of cross-lingual transfer contributes to how well a multilingual model might perform on a particular language (Jones et al., 2021). If it performs well, it may allow adequate zero-shot performance: a multilingual model trained for a task on training data taken from a single language such as English might be able to generalise its task-related knowledge to another language as well.

Previous research shows how cross-lingual transfer is enhanced by shared properties of languages. Lexical overlap, for instance, is investigated by Pfeiffer et al. (2021b). The authors aim to improve performance on languages with scripts that were unseen during the training of the multilingual language model. One of their main methods is to distinguish lexically overlapping and not overlapping vocabulary items between the target language vocabulary and the multilingual language model vocabulary. When they retrain the embedding layer of the model, they use the original multilingual language model representations of the lexically overlapping vocabulary items, only randomly

initializing representations for the rest of the vocabulary. The authors found that in the case of a token-level task such as named entity recognition, lexical initialization largely outperforms fully random initialization on languages seen during training, languages unseen during training with seen scripts, and even for languages unseen during training with unseen scripts.

Pires et al. (2019) use zero-shot transfer scenarios to evaluate the correlation between the typological similarity between languages and the amount of transfer. The authors train the model on part of speech tagging using training data from a source language, and test model performance on a target language. They find that when source and target languages share the same typological property (order of subject, object and verb and the order of adjective and noun), zero-shot performance on part of speech tagging increases compared to when the source and target languages have different values for these properties.

Wu and Dredze (2020) investigate the impact of language similarity by training bilingual BERT models training on highly related language pairs. Afrikaans is a daughter language of Dutch, and it has approximately 10 times smaller Wikipedia size. Lithuanian and Latvian are closely related languages with comparable Wikipedia sizes. Compared to monolingual BERT models trained on Afrikaans and Latvian, bilingual BERT models achieve an improvement on named entity recognition, part of speech tagging, and dependency parsing. While the improvement is not large, the results indicate that combining typologically related languages might help cross-lingual transfer. This conclusion of Wu and Dredze (2020) is an important factor in defining my own research question (see Chapter 1), and my experiments (see Chapter 4).

2.3.4 The role of inflectional morphology in language modelling difficulty

As mentioned in the introduction to this section, certain typological properties of languages seem to correlate with language modelling difficulty. Investigating how these properties impact potential language model performance may have important consequences of how these models should be trained and constructed for optimal performance. In the following, I introduce three papers, all investigating the importance of inflectional morphology for language modelling difficulty (Cotterell et al., 2018; Mielke et al., 2019; Park et al., 2021).

All three papers investigate the divergent performance of language models on different languages, testing the hypothesis whether this is due to the relative complexity of the inflectional morphology of the languages. Besides the research question, the three papers also share methodological similarities in that they all involve training open-vocabulary monolingual language models on parallel corpora. Additionally, all three papers measure language modelling difficulty in terms of surprisal. Surprisal reflects how difficult a particular segment is to model for a language model. Since longer and more complicated segments inherently have higher surprisal values, to make sure the measured surprisal values accurately reflect the success of a language model, testing needs to be done on parallel corpora.

Cotterell et al. (2018) evaluates the performance of its language models on the Europarl corpus, a parallel corpus which covers 21 European languages (Koehn, 2005). The authors measure morphological complexity by morphological counting complexity. Morphological counting complexity is calculated by enumerating the number of inflectional categories distinguished by a language. Cotterell and colleagues found that for

the languages considered, morphological counting complexity correlates with language modelling difficulty. However, the set of languages they examine is not particularly diverse, containing Indo-European languages except for three Uralic languages (Finnish, Estonian, Hungarian).

Mielke et al. (2019) extends the previous paper by considering a larger variety of languages: 69 languages from 13 distinct language families using a parallel Bible corpus (Mayer and Cysouw, 2014). Additionally, they investigate a larger number of potential factors correlating with language modelling difficulty: five typological and two statistical features. They do not find an obvious correlation between morphological complexity and language modelling difficulty. The same is true for other typological features quantifying various structural properties of languages. However, Mielke and colleagues find that measures derived from purely statistical properties of texts do show at least weak correlations with the surprisal. These measures are not independent of morphology. Raw word inventory is the number of word types, and it used as a measure of morphological complexity when normalized by the total count of words in type-token ratio. This indicates that it is perhaps not the case that the complexity of inflectional morphology does not correlate with language modelling difficulty, but that the measures in Mielke et al. (2019) are not appropriate for measuring these correlations.

A further limitation of the approach Mielke and colleagues take, however, is that most typological measures they employ are calculated based on expert judgements and NLP tools. Since adequate NLP tools are lacking for many of the low-resource languages represented in the Bible corpus (see Section 2.3.2), neither of these scores can be applied to these, only to the Europarl languages. This means that their conclusions rest on the same set of typologically similar languages Cotterell et al. (2018) used.

Park et al. (2021) set out to confirm the hypothesis that the complexity of inflectional morphology correlates with language modelling difficulty on 92 languages. As expert measures, Park and colleagues elect to use a subset of 12 different WALS features labelled with the category 'morphology' by the authors of the resource, establishing missing feature values by consulting reference grammars of the individual languages. They employ corpus-driven measures as well. These include type-token ratio (also described in Section 2.3.2) and the mean length of words, defined as the average number of characters per word token. Indeed, this allowed the authors to establish to both corpus-based and expert-driven measures of morphological complexity correlate with language modelling difficulty.

2.4 Hypotheses

In the previous sections of this chapter, I explained the basics of language modelling, how the current state-of-the-art pretrained transformer-based massively multilingual language models work, and how they are used. I also introduced adapters, a framework of language model adaptation utilizing a small number of additional parameters to expand existing language model capabilities. Finally, I described how various properties of individual languages affect the performance of multilingual language models.

Based on these theoretical foundations, I would like to explicitly state my hypotheses as answers to my research questions.

Can the stacking of language adapters contribute to increased model performance on downstream tasks? I believe stacking language adapters can in-

crease model performance by inducing cross-lingual transfer between the target and the stacked language adapters. Each adapter extends the original language model capacity with new parameters. Larger model capacity is possible to lead to better downstream performance by virtue of extending existing language model capabilities.

What extra-linguistic factors determine which language adapters combine well? While I believe stacking language adapters can increase model performance, I do not believe that this increase can happen with all language combinations. Certain language adapter combinations might have a neutral or adverse effect on downstream task performance. Extra-linguistic factors, such as the relative size of the training data of the two language adapters that I set out to combine, and whether they share their script are expected to have an influence on the success of adapter combinations. A language adapter trained on more data will likely be better trained, capable of transferring more linguistic information to the target language. Distinct scripts, on the other hand, can be a significant hindrance to the success of cross-lingual transfer.

What linguistic factors determine which language adapters combine well? I expect that certain language adapter combinations will result in better downstream performance than others. I believe that higher typological similarity between languages whose adapters are combined is one of the factors that can determine whether downstream performance increases (see Section 2.3.3 for a more detailed discussion). I also believe morphological complexity is an appropriate feature to use as a measure of typological similarity between languages due to its correlation with language modelling difficulty (see Section 2.3.4), and its ease of quantification even from raw monolingual corpora (see Section 2.3.2).

Which languages benefit the most from stacking language adapters? I predict that the stacking of language adapters will primarily benefit low resource languages, and zero-shot languages. Both language groups have relatively little training data available to train adapters with, which makes it likely that their language adapters are undertrained. Cross-lingual transfer of linguistic information from the adapter of another language might lead to a significant benefit in model knowledge in exactly these situations.

Chapter 3

Methodology

In this chapter, I provide further details on the tools that my experiments are based on. I start with describing multilingual BERT in some detail, the multilingual language model I use for my experiments. I also discuss the structure of the language adapters following Pfeiffer et al. (2020b). In the next section, I describe how I estimate morphological complexity, the feature that I use to measure the typological similarity of different languages compared to each other. Finally, I introduce the tasks I use to evaluate the performance of multilingual BERT on the individual languages, and the relevant datasets.

3.1 Multilingual BERT

In this section, I describe multilingual BERT, one of the first pretrained transformer-based massively multilingual language models. Following this, I illustrate how language adapters are added to multilingual BERT, while also touching on their structure.

3.1.1 Description of the model

Multilingual BERT was introduced by Devlin et al. (2019) in the same paper that introduced the original BERT model as well. The base BERT model and multilingual BERT are identical in terms of structure, the only differences lie in the pretraining corpus used and the size of the model vocabulary (Pires et al., 2019; Wu and Dredze, 2020).

Multilingual BERT is a multilingual language model that consists of 12 layers of transformer blocks, pretrained on the 104 largest Wikipedias. Transformer blocks employ the self-attention mechanism (Vaswani et al., 2017). In simple words, self-attention is a way for a language model to establish the relative importance of input elements with respect to each other.

Each transformer block contains 12 separate attention heads. This enables the language model in each layer to establish a variety of links between individual input elements. The hidden size of multilingual BERT, 768, corresponds to the dimensionality of the representations it creates. The model has a vocabulary size of 115,000 subword tokens, and a total of 110 million parameters (Devlin et al., 2019; Wu and Dredze, 2020). Since its introduction, other multilingual language models emerged that are larger than multilingual BERT, for instance XLM-RoBERTa (Conneau et al., 2020) with its 550 million parameters. However, larger models do not always achieve higher performance

than multilingual BERT, and this makes it a popular model even in recent work (Pfeiffer et al., 2020b; Bjerva and Augenstein, 2021). Multilingual BERT also has the largest set of pretrained language adapters available, thanks to the work of Jonas Pfeiffer and his colleagues behind AdapterHub (Pfeiffer et al., 2020a).[1]

3.1.2 Language adapters

Language adapters are a set of additional parameters added to a multilingual language model to extend its existing capabilities on a target language. Language adapters, like all other adapters, are added on top of each language model layer. Before pretraining, the original language model parameters are frozen and only the language adapter parameters are changed. Language adapters are trained with the masked language modelling training objective also described in Section 2.1.3. This involves predicting a subset of input tokens that were masked by the training algorithm (see further details in Section 2.2).

The fact that language adapters are encapsulated between frozen model parameters forces them to learn representations that are compatible with the original model. This also means that language adapters are compatible with each other and with other (task or domain) adapters, making them swappable and allowing a number of diverse configurations. Adapters can be stacked (such as in the MAD-X framework of Pfeiffer et al. (2020b)), or applied in a parallel fashion.

Stacking language and task adapters have been shown to allow a multilingual language model to benefit from both the language- and task-specific knowledge contained inside the parameters of the respective adapters. Pfeiffer et al. (2020b) demonstrate that higher performance can be achieved on named entity recognition on a series of languages when using both a language and task adapter as opposed to only using a task adapter (see Section 2.2.2). This fact shows that stacked adapters do not overwrite each other, but demonstrates that stacking adapters might be a good way to combine adapter knowledge.

Figure 3.1 illustrates how stacking adapters works inside a language model layer. The language model signal enters the current layer and traverses its transformer block (multi-head attention layer and feedforward layer). The output of this transformer block then enters the first adapter, goes through a down-projection and and an up-projection, and continues to the second adapter stacked on top of the first. The same down- and up-projection repeats there. Residual connections are maintained between the original model layer and the first adapter, and the first and second adapters, respectively.

I believe that based on this principle, we can also use stacked language adapters to explicitly encourage cross-lingual transfer. This might be especially beneficial for languages with little training data available for the purpose of pretraining with masked language modelling. By combining a language adapter of this language with a language adapter of another language with more training data, we might be able to drive cross-lingual transfer of the multilingual language model from one language to another. Since typological similarity is shown to enhance cross-lingual transfer (see Section 2.3.3 for a discussion), it is reasonable to expect that by stacking language adapters of typologically similar languages, we can achieve superior performance on a target language (see hypotheses in Section 1.1 and Section 2.4).

[1] https://adapterhub.ml

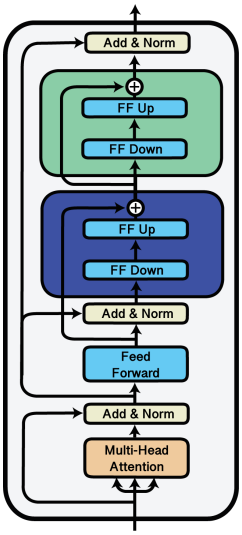


Figure 3.1: Illustration of stacked language adapters within a language model layer. The two adapters sit on top of each other, the first one receiving input from the previous language model layer, and the second one receiving input from the first adapter. The added and normalized representations (including residual connections) are then forwarded to the next language model layer. Image from `https://docs.adapterhub.ml/adapter_composition.html#stack`.

3.2 Morphological complexity

In Section 2.3.2, I showed how to quantify a number of typological features, including ones pertaining to morphological complexity. In Section 2.3.4, I introduced three papers that investigate the correlation between morphological complexity and language modelling difficulty (Cotterell et al., 2018; Mielke et al., 2019; Park et al., 2021), which shows that language models are sensitive to the morphological complexity of languages. In this section, I discuss the reasons why I picked morphological complexity as a typological feature to investigate: the fact that it affects the way language models process input and that it can be quantified using corpus-driven mesaures. I also describe how I chose to measure it.

3.2.1 Properties of morphological complexity

Pretrained transformer-based language models use a data-driven subword tokenizer to segment their input (see Section 2.3.1). The language model then assigns embedding vectors to the individual subword tokens in its embedding layer. Layer by layer, the language model adds incorporates more and more of the subword's context in its representations until it creates contextual embeddings as output. The success of the processing the language model carries out relies partially on how well the subword tokens can represent input strings of a language.

As described in Section 2.1.3, tokenizer coverage varies considerably between languages. Even in English, a high resource and language with a relatively simple morphology, subword tokens might fail to cover the same morphemes consistently depending on context. See the examples in 5 and 6, repeated from examples 3 and 4 from Section 2.3.1.

(5) a. screen *(screen)*
b. sun ##sc ##reen *(sunscreen)*

(6) a. glass ##es *(glasses)*
b. sung ##lasse ##s *(sunglasses)*

The morphemes *screen* and *glasses* are broken down into subword tokens differently, depending on whether they stand on their own (sentences 5a and 6a) or if they are parts of a compound, as in sentences 5b and 6b.

The same phenomenon is further exacerbated in languages with complex inflectional morphology where word stems are often processed in a number of different ways by the subword tokenizer, depending on the exact affixes added to a target word. Take the verb *esik* 'fall' in Hungarian. Depending on the person (1, 2, 3), number (singular or plural), aspect (imperfective or perfective) or tense (past or present), the verb stem is assigned a number of different forms by the multilingual BERT tokenizer.

In 7, I illustrate how 8 different forms of the verb *esik* 'fall' are split in Hungarian: *esem, esik, esünk, estek, elesel, estél,* and *esett*. The subword corresponding to the stem of the verb is different in each example.

(7) a. **ese** ##m
fall.1.SG.PRS.IPFV
'I fall' (imperfective)

b. **esi** ##k
fall.3.SG.PRS.IPFV
'He/she/it falls' (imperfective)

c. **es** ##ünk
fall.1.PL.PRS.IPFV
'We fall' (imperfective)

d. **este** ##k
fall.2.PL.PRS.IPFV
'You fall' (imperfective)

e. **eles** ##el
fall.2.SG.PRS.PFV
'You fall' (perfective)

f. **est** ##él
fall.2.SG.PAST.IPFV
'You fell' (imperfective)

g. **esett**
fall.3.SG.PAST.IPFV
'He/she/it fell' (imperfective)

Examples such as 5 and 6 show that subword tokens do not correspond one-to-one with morphemes even in morphologically simpler languages such as English. Learning subword representations seems to be even more difficult for morphologically complex languages such as Hungarian. This is because language models learn from distributional patterns. Words that appear in the same contexts are assumed to have a similar meaning or similar function to each other. However, when the same words are represented by a series of dissimilar subword tokens depending on various grammatical factors, such as in example 7, capturing similar distributions becomes a difficult challenge.

This difficulty in turn may affect language model knowledge on all types of downstream tasks. These include word-level tasks, such as named entity recognition or morphological probing, to sequence-level ones, like part-of-speech tagging, and to sentence-level ones, such as question-answering or machine translation.

3.2.2 Quantifying morphological complexity

Morphological complexity can be quantified in a number of different ways as also described in Section 2.3.2. Expert judgement can be used from resources such as WALS

and Universal Dependencies. WALS contains many chapters that relate to morphological complexity, including 26A "Prefixing or Suffixing in Inflectional Morphology", 22A "Inflectional Synthesis", 30A "Number of Genders", 49A "Number of Cases", and 111A "Nonperiphrastic Causative Constructions". WALS is reliable, but its drawback is that many languages are not covered by it.

Another way to characterise how complex morphological complexity of a language is to enumerate the number of inflectional categories distinguished in a language (Cotterell et al., 2018; Mielke et al., 2019). For instance, English verbs typically have five distinct forms: the root, the third person singular form, the present participle, the past form, and the past participle (see the example in 8).

(8) a. forbid *(root)*
b. forbids *(third person singular)*
c. forbidding *(present participle)*
d. forbade *(past)*
e. forbidden *(past)*

Counting these forms, however, also requires a grammar or a parser, not available for a large number of languages.

However, one of the main advantages of focusing on morphological complexity as a typological feature is that it is relatively straightforward to quantify using corpus-based measures. Such measures are described in Section 2.3.2. A simple but effective example is type-token ratio. It is calculated by taking the number of distinct word types in a text normalising it by the count of all tokens in the text. See 3.1 for the formula of type-token ratio (TTR) where V is the count of all distinct word types in the text and fr_i represents the frequency of the i^{th} word type.

$$TTR = \frac{V}{\sum_{i=1}^{V} fr_i} \tag{3.1}$$

The only preprocessing required for this is dividing into words on whitespace characters. Complex inflectional morphology results in a larger number of distinct word types, raising type-token ratio for the particular language.

However, it is important to note that type-token ratio was originally proposed as a stylometric tool, thus it is sensitive to stylistic differences between texts. Additionally, type-token ratio cannot distinguish between different word formation processes, such as inflection (9a), derivation (9b), and compounding (9c), as well as irregular (10a) and regular inflection (10b) (Bentz et al., 2016).

(9) a. eat → eats *(inflection)*
b. hope → hopeful *(derivation)*
c. fire → fireman *(compounding)*

(10) a. go → went *(irregular inflection)*
b. jump → jumped *(regular inflection)*

Despite these drawbacks, type-token ratio still excels in being a simple measure that correlates with other measures of morphological complexity (Kettunen, 2014; Bentz et al., 2016; Park et al., 2021). However, this simple formula can still be improved upon.

Park et al. (2021) use an alternative of type-token ratio called moving-average type-token ratio, first proposed by Covington and McFall (2010) for measuring lexical diversity. The moving average makes the type-token ratio calculated more resistant to effects

from text length as overall type-token ratio which can affect even parallel corpora (Covington and McFall, 2010; Kettunen, 2014). Covington and McFall (2010) recommend a moving average of 500 tokens which Park et al. (2021) also follow.

Type-token ratio or moving-average type-token ratio can straightforwardly be applied to directly compare the similarity of the morphological complexity of two languages. Lin et al. (2019) proposes the formula in 3.2 to calculate the distance (d_{ttr}) between the type-token ratios of languages a and b.

$$d_{ttr} = \left(1 - \frac{ttr_a}{ttr_b}\right)^2 \tag{3.2}$$

Let ttr_a stand for the type-token ratio of a corpus in language a, and let ttr_b stand for the type-token ratio of a corpus in The higher the morphological similarity between two languages, the closer d_{ttr} as defined in 3.2 is to 0.

3.2.3 The Bible as corpus

The success of all corpus-based measures described in Section 3.2.2 depend on the corpus they are applied on. This corpus has to be a parallel one, with the same meaning belonging to the same domain. Another advantage is if it is available for as wide a range of languages as possible.

No other books have been translated to more languages than the Bible. Originally written in Ancient Hebrew, Aramaic, and Koine Greek, its first part, the Old Testament, was first translated into Greek (so-called *Septuagint*) in the 3rd century BC. Since then it has been translated to thousands of other languages. While there are certainly differences between Bible versions (some books do not appear in some versions) the content can be largely considered parallel. This makes large collections of Bibles in different languages a massively parallel corpus, such as the Parallel Bible Corpus (Mayer and Cysouw, 2014). Mielke et al. (2019) further processed this Parallel Bible Corpus by aligning it between languages on individual Bible verses.

While the Parallel Bible Corpus covers more than 800 languages, it does not cover many smaller, low resource languages. Fortunately, there are other resources available, for instance the Parallel Bible Verses for Uralic Studies (University of Helsinki, FIN-CLARIN et al., 2020), that cover low resource languages in the Uralic family, including Komi, Udmurt, Erzya, and Khanty.[2]

These massive parallel corpora can come extremely useful in NLP research. Mielke et al. (2019) and Park et al. (2021), for instance, use Bible excerpts to evaluate how well language models trained on monolingual data can represent the given language. Since the Bible excerpts have parallel content, this is expected to make the results comparable between languages (see Section 2.3.4 for further details on their methods and results).

Bible corpora are excellent resources to generate comparable type-token ratio values, described in the previous section. Bentz et al. (2016) uses the full text of the Parallel Bible Corpus (Mayer and Cysouw, 2014) to calculate type-token ratio. Park et al. (2021) uses a subset of the same corpus for calculating both a simple type-token ratio and a moving-average type-token ratio.

[2] https://metashare.csc.fi/repository/browse/parallel-bible-verses-for-uralic-studies-korp/f2a92fbe5e2111ea8e93005056be118ed58a9238ff0e4382bb3ade965f67e4f9/

The parallel content and the comparable domain (religious text) of the Bible translations ensures a certain stylistic similarity between versions in different languages that allows approximating morphological complexity using type-token ratio. However, there are certain caveats that have to be made when using the Bible as a parallel corpus.

While typically the language of the Bible is archaic and not particularly representative of modern language use, in many languages more recent Bible versions exist with a more contemporary style. A more contemporary style, however, does not make the domain of the text any closer to modern language use. Additionally, since these 'simple' translations are not available for all languages, there is potential danger in comparing Bible versions with more traditional and with more modern style. This might make it more difficult to compare the type-token ratio between different languages.

Additionally, virtually all Bible versions are translations. Translated texts often end up with interference from the source language, making them more removed from everyday language use (Mielke et al., 2019).

3.3 Evaluation

Multilingual benchmarks such as XTREME (Hu et al., 2020) and XTREME-R (Ruder et al., 2021) provide a way to evaluate the degree of linguistic knowledge multilingual language models might have on different languages. These benchmarks also help determining the capabilities of the multilingual language model of doing zero-shot cross-lingual transfer since the training dataset is typically given in English with evaluation datasets in a varying number of other languages.

XTREME covers 40 languages and contains 9 tasks (Hu et al., 2020). The tasks differ in the amount of syntactic and semantic information they require, and include token-level sequence classification, such as part-of-speech tagging and named entity recognition, reasoning tasks such as cross-lingual natural language inference, and retrieval tasks such as parallel text mining, and cross-lingual sentence retrieval (see Section 2.1.2 on a discussion on these latter two tasks).

XTREME-R is a revision of the original XTREME benchmark, covering a larger set of languages, 50, over 10 tasks. XTREME-R aims to have more challenging tasks testing a typologically more diverse set of languages. It introduces new tasks and abandons tasks from XTREME that are perceived to either not have the typological coverage, or that are not challenging enough for existing models.

Both multilingual benchmarks use the cross-lingual sentence retrieval (also called translation pair detection) task called Tatoeba. The test set on this task covers the largest set of languages, which allows testing on low resource languages. In the following section, I describe this task, also introducing the source of data I use in my experiments.

3.3.1 Cross-lingual sentence retrieval

Cross-lingual capabilities of multilingual language models are based on the quality of the joint embedding space different languages are mapped to (see Section 2.1.2). This means representations of various languages inside such a multilingual model should align with each other. This compatibility can be tested in a number of different ways, including directly measuring the alignment and isomorphism of the embedding spaces of various languages (Jones et al., 2021), testing zero-shot performance, or finding appropriate downstream tasks, such as cross-lingual sentence retrieval.

Cross-lingual sentence retrieval is a similarity search task between sentences in a cross-lingual setup. Alternatively, it can also be interpreted as a ranking task. Given a dataset of parallel sentence pairs in two different languages, the task involves finding candidate translations for the individual sentences in one of the languages. This is done by using the language model to create sentence-level representations. In the case of multilingual BERT, this is done by averaging the subword token representations of the output layer. Afterwards, candidate translations are identified using a distance measure, typically cosine similarity, between these representations. If performance of the language model is high on cross-lingual sentence retrieval, that means that it is able to create well-aligned representations across languages, showing robust cross-lingual knowledge.

The most well-known variant of the task is Tatoeba, adopted from the open collection of English sentences and translations maintained by an online community of volunteers.[3] The Tatoeba task was proposed by Artetxe and Schwenk (2019) as a way to test multilingual language model capabilities in a task with a higher coverage than other benchmark tasks such as cross-lingual natural language inference (XNLI), cross-lingual document classification (MLDoc), and parallel corpus mining (BUCC). It became adopted in both the XTREME and XTREME-R benchmarks (Hu et al., 2020; Ruder et al., 2021). The way Artetxe and Schwenk (2019) formulate the task is by encoding sentences using a multilingual language model, finding the nearest neighbour to each sentence using cosine similarity, and calculating the error rate. Evaluation takes place by calculating the error rate, i.e., the accuracy score on the task.

Cross-lingual sentence retrieval, as an evaluation task, has advantages besides the comparatively good coverage. First, as it is formulated in XTREME and XTREME-R, it does not involve task-specific adaptation. The multilingual language model outputs are used directly as representations of the sentence elements, and matching sentences are searched using cosine similarity. This allows a more immediate testing of the added benefits of language adapters, and introduces no further bias from effects such as the source language of the training data.

Additionally, the company TAUS that I carry out my internship with has both data and products that are relevant to cross-lingual sentence retrieval. TAUS provides data and services related to neural machine translation. They have large corpora of aligned sentence pairs across a broad range of different languages (e.g., TAUS Data Cloud). Comparing cross-lingual sentence representations is already part of the data cleaning flow of the company. Moreover, another TAUS service, Matching Data, involves retrieving sentences from their corpora that match the domain of a target set. This means that any improvements achieved on the task can be useful for the company itself.

Additionally, there are other datasets available that help adapt the task to an even wider range of languages (see next section).

3.3.2 Datasets

Cross-lingual sentence retrieval is a task closely related to machine translation. The parallel datasets that are required to train neural machine translators are also useful for testing the cross-lingual sentence retrieval capabilities of language models. This means that besides the Tatoeba dataset assembled by Artetxe and Schwenk (2019), there are many other sources that can be used as test sets to allow a wider coverage of

[3] https://tatoeba.org/en/

languages.

Tatoeba dataset

As mentioned before, the cross-lingual sentence retrieval task using the sentences from the Tatoeba collection was proposed by Artetxe and Schwenk (2019). The authors retrieved a snapshot of the dataset, carried out an automatic cleaning process by removing sentences containing elements such as @ and http, removing sentences with fewer than three words, and duplicate sentences. This resulted in up to 1,000 aligned sentences between English and a set of 72 languages, and a total of 112 languages with at least 100 parallel sentences with English.

A smaller set of 33 and 38 languages, respectively, are adopted by the XTREME and XTREME-R benchmarks retaining the original formulation of the task (Hu et al., 2020; Ruder et al., 2021).

Tatoeba Challenge dataset

Tiedemann (2020) proposes the Tatoeba Challenge or Tatoeba Translation Challenge dataset, a so-called 'realistic' dataset for evaluating low resource and multilingual machine translation.[4] It is considerably larger than the Tatoeba set put forward by Artetxe and Schwenk (2019), containing 500GB of compressed data of 555 languages arranged in 2961 distinct language pairs. Most language pairs have both training and test datasets. The training data is taken from OPUS[5], an open collection of online parallel corpora (Tiedemann, 2012), and the test data taken from the Tatoeba collection (also used by Artetxe and Schwenk (2019)).

Tatoeba Challenge was created out of pre-aligned parallel texts that were further cleaned for non-printable and non-Unicode characters, also de-escaping special characters. Automatic language identification was applied to verify whether the sentences truly belong to the source and target languages.

Since it was originally created for training and evaluating neural machine translation solutions, the format of the Tatoeba Challenge dataset does not directly match the format of the original Tatoeba dataset. For instance, data for different languages is split into training, test, and development sets. Additionally, parallel data is available in a large number of language pairs and English is not always included in the source or target side.

TAUS Data Cloud

TAUS Data Cloud is a resource of translation datasets covering sentence pairs from 56 languages across various domains and content types. The Data Cloud as a product is used for training and testing machine translation engines as well as improving the performance of existing machine translation solutions.

The source of the TAUS Data Cloud sentences is the translation memory of computer-assisted translation tools that professional translators use to aid their work. Translation memory allows the retention of source and target language pairs, which can speed up the translator's work by recalling previous instances of how the same words, phrases,

[4] https://github.com/Helsinki-NLP/Tatoeba-Challenge
[5] https://opus.nlpl.eu

or sentences were translated before. This translation memory can be exported from computer-assisted translation tools, serving as a high quality parallel corpus.

Similarly to the Tatoeba (Artetxe and Schwenk, 2019) and Tatoeba Challenge (Tiedemann, 2020) datasets, the TAUS Data Cloud also goes through an automatic cleaning process. This process is somewhat more involved than what is done with the other two corpora. The aim of the cleaning is among all to remove poor and imprecise translations, misaligned sentences, duplication errors, and correct wrong encodings.

The main fixes the TAUS data cleaner carries out involve fixing broken encodings, extracting content between HTML tags, fixing common spelling mistakes and typos, such as "sofware" for "software", and replacing characters from the wrong script with correct ones.

Afterwards, a series of filters is applied to filter out false translations and noise. These aim to remove duplicate sentences, filter out sentence pairs with a significant length difference, and verifying whether the sentence is in the correct language.

Following this cleaning process, a cleaning stage analogous to the task of cross-lingual sentence retrieval is commenced. Sentence embeddings are created of the source and target sentences using sentence embedding model, such as LASER (Artetxe and Schwenk, 2019) or LaBSE (Feng et al., 2021). The two sentences are only accepted as true translations if they both pass the fixes and filters of the cleaning process, and if the cosine similarity of their embeddings exceeds a certain threshold.

3.4 Summary

In this chapter, I provided details on the methods used to address the research questions of this thesis. In Section 3.1, I described multilingual BERT, the language model I use in my experiments, and showed how language adapters and stacked language adapters may be injected between the model layers.

In Section 3.2, I reasoned for selecting morphological complexity, the feature I use to measure typological similarity between languages in my experiments. I also defined overall type-token ratio and moving average type-token ratio as corpus-based measures of morphological complexity, how I used them to establish typological similarity between languages, and also showed how I calculate them on a parallel corpus.

Finally, in Section 3.3, I introduced *cross-lingual sentence retrieval* or translation pair detection as a cross-lingual task evaluating language model knowledge on specific languages. I outlined how the task is formulated following multilingual benchmarks. I also described datasets that can be used as evaluation sets for the cross-lingual sentence retrieval task.

Chapter 4

Experimental Design

In the previous chapters, I described the background of my work and the methods I use to test my hypotheses. In my thesis, I aim to discover whether cross-lingual transfer can be induced in multilingual BERT using stacked language adapters. This chapter describes how I prepared the experiments to address the hypotheses related to this question that were described in Chapter 1 and Chapter 2.

I start with reiterating my hypotheses and how I operationalise them. Then, I describe how I assemble the evaluation sets and the parallel corpus used to measure morphological complexity. I also discuss how I arrived at the final selection of languages. Finally, I introduce the different experimental setups. Since there are a large number of experiments, I explain the experiments using examples for a single language.

4.1 Operationalisation of research questions

In Chapter 1, I described the main goal of my thesis, i.e., exploiting the effects of typological similarity between different languages to increase multilingual language model knowledge on low resource languages. In this section, I reiterate my reseach questions, also briefly introducing the methods that I use to test them.

- Can the stacking of language adapters contribute to increased model performance on downstream tasks?
- Which linguistic factors determine which language adapters are useful to combine?
    1. Is typological similarity between the languages a good predictor for which language adapters combine well?
    2. Are simple corpus-based measures such as type-token ratio good estimators for morphological complexity of particular languages?
    3. Does similarity in terms of morphological complexity well represent typological similarity between languages?
- Which extra-linguistic factors determine which language adapters are useful to combine?
    1. Do training-related factors, such as the size of the training data for the two language adapters determine the contribution of language adapter stacking to the success on downstream tasks?

2. Do other factors related to the training data, such as shared scripts, determine the contribution of language adapter stacking to the success on downstream tasks?

- Which languages benefit the most from stacking language adapters?

1. Does model performance increase on all languages or only on low resource languages?
2. Does model performance increase on zero-shot languages, i.e., languages that the language model has not seen during its pretraining?

These research questions all serve to test my main hypothesis. I hypothesise that language model knowledge on specific languages can be enhanced by exploiting cross-lingual transfer of linguistic information between languages. I predict this effect to be strongest when the languages share a high enough degree of similarity in terms of typological features, and when the target language is a low resource language. Moreover, I assume that typological similarity between languages can be measured using morphological complexity measures.

The technique I chose to address these research questions is to encourage cross-lingual transfer using language adapters, bottleneck modules of additional model parameters that serve to extend existing language model capabilities on the target language (see Section 2.2 and Section 3.1.2 on further details on adapters).

4.2 Preparing experiments

In the following section, I describe the steps I need to prepare the experiments that aim to address my research questions. I first describe how I measure the morphological complexity of different languages, as well as how I compare these measures with each other. Then, I discuss how I arrived at the final selection of languages that I used in my experiments. Finally, I describe how I prepared the data to evaluate the success of different experiments on cross-lingual sentence retrieval.

4.2.1 Measuring morphological complexity

As described in Section 3.2.2, I use type-token ratio as a proxy for establishing the morphological complexity of various languages. In practice, there are two closely related measures I employ, ordinary type-token ratio, and a moving average type-token ratio. Following Park et al. (2021), I calculate both of these measures on a parallel corpus of the New Testament for each of the 50 languages for which pretrained multilingual BERT language adapters were available on AdapterHub (Pfeiffer et al., 2020a), using the Python module *LexicalRichness*.[1] In calculating the moving average type-token ratio, I set the window size to 500 words, following Park et al. (2021) and Covington and McFall (2010).

To establish the degree of distance with respect to these two scores of the different languages, I use the following formula in 4.1 from Lin et al. (2019), repeated from 3.2, also in Section 3.2.2.

[1] https://github.com/LSYS/LexicalRichness

$$d_{ttr} = \left(1 - \frac{ttr_a}{ttr_b}\right)^2 \tag{4.1}$$

In 4.1, *ttr* stands for either of the morphological complexity measures mentioned in the previous paragraph, measured for either language a or b. The formula predicts that the closer d_{ttr}, i.e., the distance in terms of type-token ratio, is to 0, the more similar the two languages are in terms of morphological complexity. Following 4.1, languages with language adapters can be ranked with respect to morphological similarity to the target language.

The source of the parallel corpus is the Parallel Bible Corpus by Mayer and Cysouw (2014), also introduced in Section 3.2.3.[2] The corpus is already tokenized, which is important in order to avoid inflating the number of types present in the text (words with punctuation attached to them would register as distinct word forms). However, there are a few challenges I needed to deal with before calculating morphological complexity for the individual languages.

First, while some of the Bible texts contain both the Old and New Testament, a large subset of languages only had New Testament texts available. This meant I had to limit myself to the New Testament to ensure parallel content across languages. Since the Parallel Bible Corpus is not annotated for subdivisions of the Bible, this meant that I had to identify the first line of the New Testament (see in (11)), and find the corresponding line in all Bible versions:

(11) This is the genealogy of Jesus the Messiah the son of David, the son of Abraham:[3]

For most languages, especially ones written with the Latin, Greek, and Cyrillic scripts, the line was straightforward to identify. For other languages, I used machine translation resources such as Google Translate[4] and Yandex Translate[5].

Another challenge with the Parallel Bible Corpus stems from the fact that Bible translations are not equal in terms of style. While modern translations tend to be translated in a more colloquial language, a lot of Bible versions are archaic or purposefully archaising in style. Since type-token ratio was invented as a measure of style and lexical diversity, it is meant to be sensitive to this variation. This is a difficulty in applying type-token ratio as a measure of morphological complexity.

Mielke et al. (2019) show that individual Bible versions in the same language do not significantly deviate from each other in terms of language modelling difficulty. Also, when averaging the morphological complexity scores for the various Bible versions in the same language, variance stays low, in the range of the 3rd decimal place or lower (see Table B.2 in Appendix B for details). This indicates that stylistic differences between Bible versions might not make such a considerable difference.

Ideally, we would need to make a selection of Bible versions that are consistent in style across languages. However, the Parallel Bible Corpus lacks metadata on the style and date of the various translations. This makes it difficult to make a meaningful selection across Bible versions for many languages that might have 5-6 alternative

[2] Special thanks to Sabrina J. Mielke for providing me with the dataset that she also used in Mielke et al. (2019). The dataset was not available on the original location.

[3] Source: https://www.biblestudytools.com/matthew/1.html

[4] https://translate.google.com

[5] https://translate.yandex.com

texts, and especially for high resource languages, such as English and German, that boast more than 20 different versions. However, for most languages, especially for low resource ones, there is typically only a single Bible version is available. This means we cannot pick between Bible versions, we have to use the version we have access to.

Table 4.1 shows line counts averaged over Bible versions of a selection of languages, demonstrating that at least in terms of content length, the resulting corpora are mostly parallel with each other.

To get language-level scores, I average the overall type-token ratio and moving average type-token ratio values across Bible versions of the language. Table 4.2 shows the resulting morphological complexity scores for a selection of languages.[6] Afterwards, the distance between languages is measured using the formula in 4.1.

The lowest distance scores for both the overall type-token ratio and the moving average type-token ratio seems to depend on the language. This diversity, however, does not seem to manifest when the highest distance scores are measured. For each language the highest distance score measured in overall type-token ratio is Vietnamese (except of course for Vietnamese). On the other hand, the highest distance score measured in moving average type-token ratio is either English (for more morphologically complex languages), or Korean (for less morphologically complex ones).[7] This is likely due to the relative extremity of the languages in the degree of morphological complexity. Vietnamese is a language with an isolating morphology where both syllables and words are separated using whitespace characters, and English is also not morphologically complex. On the other hand, Korean has a complex agglutinative morphology.

Before the section ends, I have to mention Haitian Creole, a language I decided to exclude from my experiments.

Haitian Creole is a creole language, the only one that has a pretrained language adapter on AdapterHub. Before its exclusion from the sample, it had the highest distance score compared to all other languages using either overall type-token ratio or moving average type-token ratio. This can probably be explained with the fact that creole languages are supposed to have a set of features that differentiate them from other languages, which include the lack of inflectional morphology and a robust auxiliary system (Bickerton, 1984). These contribute to low type-token ratios for the language.

However, I decided to exclude Haitian Creole both to ensure a higher degree of variety in the language adapters used in the experiments, and because Haitian Creole is a zero-shot language. This fact makes it unlikely that it could contribute to successful cross-lingual transfer.

4.2.2 Preparing for the cross-lingual sentence retrieval task

The data that I can use for cross-lingual sentence retrieval comes from three sources, also detailed in Section 3.3.1. One of these is a subset of the Tatoeba task dataset (Artetxe and Schwenk, 2019) in XTREME (Hu et al., 2020) and XTREME-R (Ruder et al., 2021), two multilingual benchmarks, covering 38 languages. The second source is the TAUS Data Cloud provided by the company TAUS, covering 56 different languages. And the final source is the Tatoeba Challenge dataset (Tiedemann, 2020) covering 557 languages.

[6] See Table B.2 in Appendix B for the scores for languages with pretrained language adapters.
[7] See Table B.3 in Appendix B for the relations between languages with pretrained language adapters.

Language	Number of Bible versions	Average line count
Amharic	1	7656
English	27	7941
Erzya	1	7955
French	16	8172
*Georgian	1	4894
German	22	8310
Hungarian	5	7930
Icelandic	1	7860
Indonesian	5	7936
Meadow Mari	1	7946
Northern Sami	1	7951
Turkish	3	7442
Vietnamese	6	7903

Table 4.1: Line counts of New Testament texts averaged over all available Bible versions in the Parallel Bible Corpus for a selection of languages. Average line counts are comparable across the board, except for Georgian (marked with an asterisk *).

Language	Average TTR score	Average MATTR score
Amharic	.212	.628
English	.034	.410
Erzya	.117	.608
French	.055	.453
German	.061	.484
Hungarian	.144	.567
Icelandic	.084	.507
Indonesian	.042	.445
Meadow Mari	.104	.606
Northern Sami	.097	.540
Turkish	.169	.662
Vietnamese	.017	.422

Table 4.2: Morphological complexity scores averaged over a selection of languages in the Parallel Bible Corpus. TTR stands for overall type-token ratio and MATTR stands for moving average type-token ratio.

When possible, I decided to use data from the Tatoeba task dataset (Artetxe and Schwenk, 2019). This allows better comparison with other experiments using the XTREME and XTREME-R benchmarks. If the language is not covered by this set, I consult the TAUS Data Cloud. If neither resources cover a particular language, I include data from the Tatoeba Challenge dataset which is the most extensive but less reliable in terms of quality (Tiedemann, 2020).

Since evaluation sets in the Tatoeba dataset (Artetxe and Schwenk, 2019) contain 1,000 sentence pairs between the target language and English, I made sure to also select 1,000 sentences from the other resources as well. For the Tatoeba Challenge dataset (Tiedemann, 2020), the test split is often shorter than 1,000 sentence pairs, especially when the target language is low resource. Since the definition of the cross-lingual sentence retrieval task in the XTREME and XTREME-R benchmark does not require training, in case the test set was shorter than 1,000 sentence pairs, I added sentences first from the development, then the training splits, until a final set of 1,000 sentence pairs is reached.[8]

As a final step before the experiments, I also carried out a quality check of test sets for each language, regardless of their source. I randomly sampled 50–100 sentence pairs for each language, i.e., 5-10% of all data, and I checked whether the majority of the sentence pairs are at least acceptable translations of each other. While translation quality is not even across languages, it was only the Burmese data that was sufficiently low quality so I decided to withhold the language from further testing.

4.2.3 Selecting languages

In selecting languages to use in the experiments, the goal is to arrive at a diverse sample in terms of both typology and genealogy, covering both low and high resource languages. It is also important for the language to have a pretrained language adapter, which makes it possible to carry out the experiments without needing to train additional adapters. AdapterHub (Pfeiffer et al., 2020a) provides pretrained language adapters for a large set of languages that are compatible with the multilingual BERT model.

To be able to compare the language against other languages, it is also necessary that at least one Bible version in the language is included in the Parallel Bible Corpus (Mayer and Cysouw, 2014), also mentioned in the previous section. Most of the languages with adapters on AdapterHub also have Bible versions in this corpus, with a few exceptions, such as Chinese and Cantonese.

Finally, an evaluation dataset is needed that allows measuring language model performance on the cross-lingual sentence retrieval task. Fortunately, the datasets available for this task have a sufficiently wide coverage to accommodate all languages also covered by AdapterHub and the Parallel Bible Corpus (see details on the datasets in Section 3.3.2).[9]

In the end, I arrived at an evaluation set containing eight languages and English. This set covers languages from five language families with diverse typological features

[8] For a small subset of languages, there is a total of less than 1,000 sentences in the training, development, and test splits combined. Languages like these, such as Erzya, are thus excluded from the final test set.

[9] The table on Figure B.1 in Appendix B shows the full range of factors that I considered as assembling the final evaluation set of languages, including language family, the presence of pretrained adapters, the amount of training corpora available, and the coverage of three benchmarks, XGLUE (Liang et al., 2020) and the already mentioned XTREME (Hu et al., 2020) and XTREME-R (Ruder et al., 2021).

and three different scripts. Three of the languages is zero-shot, Amharic, Meadow Mari, and Northern Sami. An additional language, Icelandic, is low resource, while Hungarian, Indonesian and Vietnamese are moderately high resource languages. French, is a high resource language. Another high resource language, English, is used as one half of the sentence pairs for the cross-lingual sentence retrieval task.

In the remainder of this section, I attempt to briefly introduce these languages. First, language-specific details are required to interpret results, but also because unique sets of diverse characteristics of the different languages should be recognised, not just reduced to the level of individual data points. Figure 4.1 also shows the first lines in the Book of Matthew for each language, selected from a random Bible edition for each language.

Language	First sentence
Amharic	የዳዊት ልጅ የአብርሃም ልጅ የኢየሱስ ክርስቶስ ትውልድ መጽሐፍ ።
Meadow Mari	Авраам тукым Эргын , Давид тукым Эргын Иисус Христосын тукымвожшо :
English	A record of the ancestors of Jesus Christ , son of David , son of Abraham :
French	Voici la liste des ancêtres de Jésus-Christ , descendant de David , lui-même descendant d'Abraham .
Hungarian	Ez Jézus Krisztus családfája . Jézus Dávid családjából származik , Dávid pedig Ábrahám családjából .
Indonesian	Inilah daftar nenek moyang Yesus Kristus , keturunan Daud , keturunan Abraham .
Icelandic	Ættartala Jesú Krists , sonar Davíðs , sonar Abrahams .
Northern Sami	Dát lea Jesus Kristusa , Dávveda bártni ja Abrahama bártni , sohka .
Vietnamese	Gia phả của Đức Chúa Jêsus Christ , con cháu Đa-vít , con cháu Áp-ra-ham .

Figure 4.1: The first line of the Book of Matthew for the languages in my evaluation sample and English.

English is a West Germanic Indo-European language spoken worldwide. English inflectional morphology is relatively simple with only a handful of different affixes. It is written with the Latin alphabet, and it has the most training corpora available out of all languages. In cross-lingual sentence retrieval experiments, sentences in different languages are matched with candidate translations in English (Artetxe and Schwenk, 2019; Hu et al., 2020; Ruder et al., 2021). Additionally, multilingual language model performance is typically the highest in English across a variety of tasks (Wu and Dredze, 2019; Conneau et al., 2020). This implies that if a language model is capable of creating well-aligned embeddings on a target language with English embeddings, its knowledge on the target language should be comparably high.

Amharic is an Ethiopian language belonging to the Semitic branch of the Afroasi-

atic language family. Similarly to other Semitic languages, such as Arabic, Amharic has a templatic morphology based on consonant roots. Amharic is written with the Ge'ez script, an alphasyllabary. Both the language and its script are zero-shot, i.e., neither were included in the pretraining data for the multilingual BERT model (Pfeiffer et al., 2021b).

Meadow Mari is a Uralic language spoken in Russia. It is one of the three standard forms of Mari. The morphology of Meadow Mari, like other Uralic languages, is primarily agglutinative. The language is written with the Cyrillic alphabet, and it is a zero-shot language just like Amharic.

French is in the Romance branch of the Indo-European language family, spoken worldwide. French, especially written French, has a somewhat more complex morphology than English. It is written with the Latin alphabet, and it is one of the few truly high resource languages in NLP.

Hungarian is a Uralic language spoken primarily in East-Central Europe. Hungarian morphology is agglutinative. It is written using the Latin alphabet, and it is a moderately high resource language.

Indonesian is an Austronesian language belonging to the Malayo-Polenesian branch of the language family. Its morphology is 'mildly' agglutinative (Moeljadi et al., 2015). It is written with the Latin alphabet, and it is a moderately high resource language, similarly to Hungarian.

Icelandic is a North Germanic language in the Indo-European language family with a more complex inflectional morphology than French or especially English. It is written with the Latin alphabet, and while Icelandic corpora are included in the pretraining data of multilingual BERT, it is considered to be a low resource language.

Northern Sami is a Uralic language spoken in Scandinavia with an agglutinative morphology. It is written with the Latin alphabet, and it is a zero-shot language.

Vietnamese is the final language in the set of languages I use in my experiments. It is in the Vietic branch of the Austroasiatic language family with an analytic language with comparatively little morphology. It is written with the Latin alphabet, and it is a moderately high resource language.

4.3 Experiments

In this section, I introduce the experiments with cross-lingual sentence retrieval that I carry out for each of the eight test languages described in the previous section to address the research questions of this thesis. My experiments have a number of common characteristics that are worth summarising.

First, the multilingual language model I use in all of my tests is multilingual BERT (Devlin et al., 2019), described in more detail in Chapter 3. Multilingual BERT creates contextual embeddings on the level of a subtoken and not on the level of a sentence. The CLS token that languages models such as multilingual BERT prepend to the beginning of each input sentence is often used as input to sentence-level tasks (Devlin et al., 2019). However, CLS token embeddings as sentence representations do not perform well on cross-lingual retrieval tasks such as the task I use (Hu et al., 2020). They hypothesise this might be due to the fact that CLS tokens might contain too high-level, semantic information, not preserving enough information on the level of the token to connect true translations across languages. An alternative way to create sentence-level representations is to average subtoken embeddings that Hu et al. (2020) show to

outperform using the CLS token. For the sake of simplicity, I average the subtoken embeddings of the output layer of the model. Although using the middle layer of the models might result in superior performance (Hu et al., 2020), there is precedent to using the final layer (Phang et al., 2020) too.

The second common feature in all experiments is that all tasks involve identifying translation pairs between English and target languages, using cosine similarity of the sentence-level representations to find the nearest neighbours across languages. In this, my experiments follow those included in multilingual benchmarks such as XTREME (Hu et al., 2020) and XTREME-R (Ruder et al., 2021). By virtue of trying to identify translation pairs between English and target languages, I am implicitly comparing the compatibility of representations between English and the target languages.

This cross-lingual sentence retrieval task allows setting up experiments across a series of languages otherwise not covered by evaluation datasets. This enables us to observe main trends across languages, revealing which adapter combinations might benefit performance, thus addressing the research questions. The drawback is that the test data is not parallel across all languages, meaning that direct comparisons of accuracy scores between languages is not possible.

If language model representations of another language turn out to be compatible with the representations of English, that might be a strong indicator of high quality model knowledge on this other language as well. Since English has disproportionately more training data available than all other languages, and generally better performance on downstream tasks than other languages, language model knowledge might be the highest quality for English (Wu and Dredze, 2019; Conneau et al., 2020).

Another advantage when comparing with English is that perhaps the most common scenario in neural machine translation in a commercial setting is translating between English and other languages. A multilingual language model can used to clean and prepare parallel bilingual data for improving machine translation solutions, just like TAUS does, and it is most likely that at least one member of the language pair considered would be English.

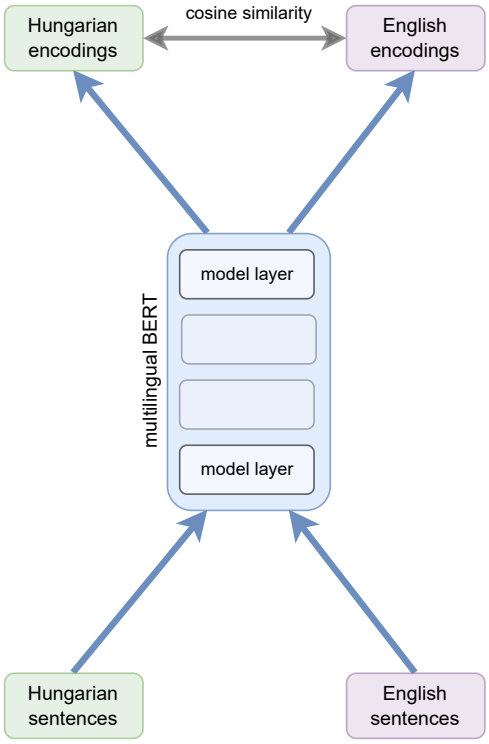
Due to the large number of experiments that I carried out, in the following I focus on making the experimental setups more legible by exemplifying them using a single target language, Hungarian. Full discussion of the results will follow in Chapter 5.

All experiments were carried out using virtual machines accessed via *Google Colab* [10]. The implementation of the experimental setup follows the implementation of the XTREME and XTREME-R benchmarks.[11] Evaluation is done by calculating error rate, i.e., accuracy, in returning the correct translation in the first position ($k = 1$) following practice in XTREME and XTREME-R. I also calculate accuracy on the $k = 3$ setup when the correct translation has to be among the top three candidate translations returned.

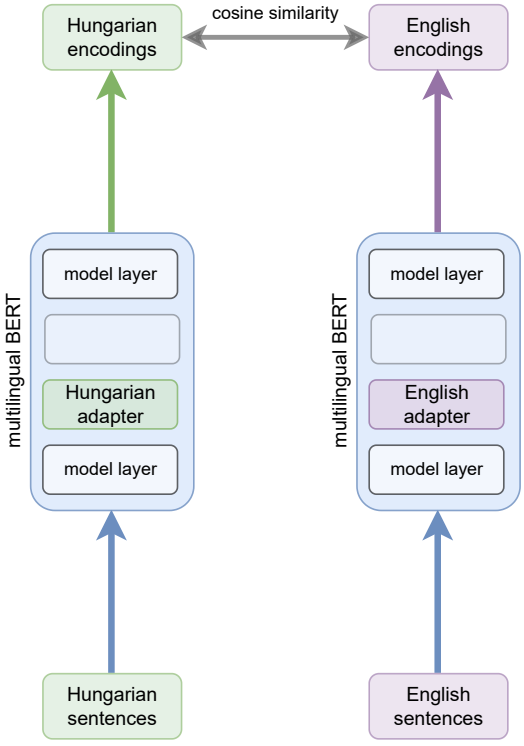
4.3.1 Baselines

Baseline experiments allow assessing the performance benefits of adapter stacks. I carried out three baseline experiments: an experiment when no language adapters were used (I name it **no-lang**), an experiment using only the target language adapter

[10] `https://colab.research.google.com`
[11] Shared GitHub page: `https://github.com/google-research/xtreme`



(a) **no-lang** baseline experiment



(b) **target-lang** baseline experiment

Figure 4.2: Diagrams representing the flow of the baseline experiments. In the **no-lang** baseline, both English and Hungarian sentences are encoded using the multilingual BERT model without adapters (see Figure 4.2a), while in the **target-lang** baseline, sentences are encoded using the relevant language adapter (see Figure 4.2b). Resulting sentence representations are then used in the cross-lingual sentence retrieval task to identify translation pairs in using cosine similarity.

(**target-lang**), and a random baseline to assess the overall difficulty of the task. See Figure 4.2 for an illustration of the baseline experiments.

In the first baseline, I only use the non-adapted multilingual BERT model to create sentence-level representations for both English and the target language. In the following discussions, I label this experimental setup **no-lang** (*no language adapters*). This measures the ability of the pretrained language model to create cross-lingually aligning sentence representations between English and different target languages. If the use of adapters results in a worse performance than when no language adapters are used, that indicates that the additional model parameters in the adapters actively hurt alignment between different languages inside the language model. For Hungarian, this baseline includes encoding both English and Hungarian sentences with multilingual BERT without adapters, and finding nearest neighbours between Hungarian and English sentences using cosine similarity.

The next baseline involves adding the language adapter of the target language on multilingual BERT before encoding the sentences of the various languages. I call this experimental setup **target-lang** after the phrase *target language adapter*. I encode English sentences using the English language adapter with multilingual BERT. Afterwards, I encode Hungarian sentences using a Hungarian language adapter, replacing the English language adapter in the model layers.

This setup allows assessing the contribution language adapters provide to creating well-aligned cross-lingual representations. The graph in Figure 4.3 to show this is indeed the case for Hungarian. Additionally, adapter stacks can directly be compared against this **target-lang** setup. Superior model performance compared to **target-lang** might indicate that model knowledge benefits from the cross-lingual transfer between the target language adapter and the stacked language adapter. Conversely, if performance is harmed by a stacked language adapter compared to the **target-lang** baseline, that might indicate that the stacked language adapter transfers irrelevant or incorrect information, obscuring the original information added by the target language adapter.

If my hypotheses stand, various language adapter combinations where the languages are similar in terms of morphological complexity should outperform both of these baselines.

I also use a **random baseline**. It serves as an indication of the task difficulty. If performance with the random baseline is too high, the task is too simplistic to measure the success of different models or experimental setups. The random baseline involves assigning a random candidate translation for each sentence in the source set. The full test set contains 1,000 sentence pairs, and random baseline does not result in higher than 0.4% accuracy scores in selecting the correct sentences. All other results are considerably higher than this accuracy. Thus we can conclude that the cross-lingual sentence retrieval task in this thesis is challenging enough to allow adequate comparison of experimental setups.

4.3.2 Adapter stacks

In the previous section, I discussed the baselines used. In this section, I discuss the adapter stacks that I use to address my hypotheses: first, I assume some language adapter combinations will be more successful than only using the target language adapter for a particular language. Second, I assume that the most successful adapter combinations will be those that involve languages with comparably complex morphology. Third, I assume that any potential positive effects of combined adapters will be

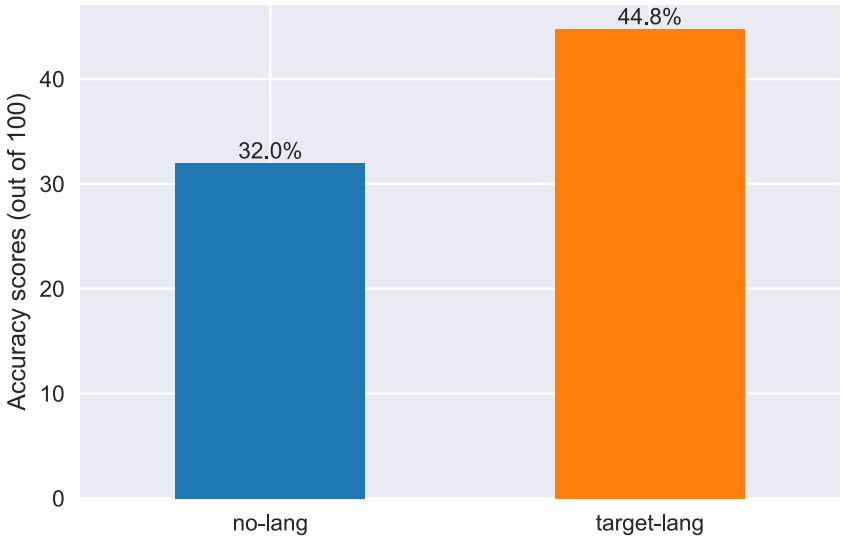


Figure 4.3: Comparing accuracy yielded by the baselines of **no-lang** and **target-lang** on cross-lingual sentence retrieval between Hungarian and English sentences. Performance is higher on the **target-lang** baseline when the Hungarian language adapter is used.

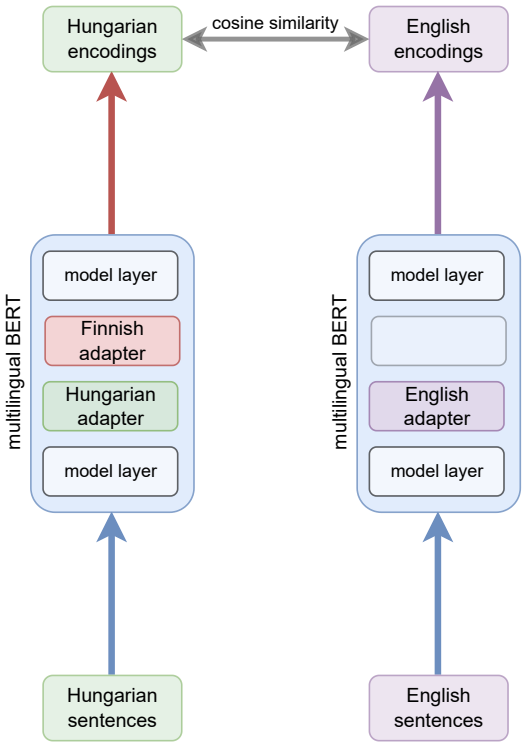
most significant when the target language is a low resource one.

As described in Section 3.2.2, I use overall type-token ratio and moving average type-token ratio scores to measure and compare the morphological complexity of various languages. In the following, I describe the setups involving adapter stacks. See Figure 4.4 for an illustration of the setups involving the overall type-token ratio. The principle shown in the illustration is the same when the moving average type-token ratio is used instead.

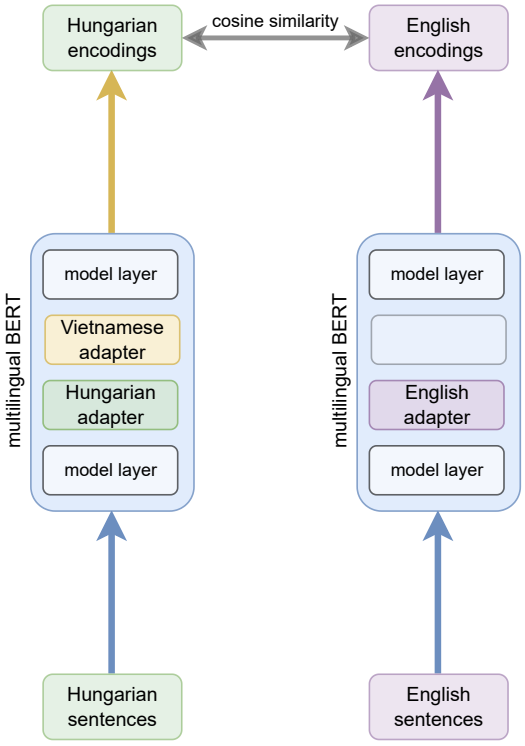
Adapter stacks based on overall type-token ratio

For each language, the first two adapter stacks are set up based on the overall type-token ratio. The **low-dttr** experiment involves a pairing with a language that is similar to the target language in terms of morphological complexity, and the **high-dttr** experiment is the exact opposite, the furthest in terms of this measure. In both cases, the adapter stack is used to encode the Hungarian sentences, and this is compared against the English sentences that were encoded using the English language adapter. The cross-lingual sentence retrieval task is carried out the same way as with the baseline experiments. For Hungarian, these two languages are Finnish and Vietnamese, respectively. Since both Finnish and Hungarian are agglutinative languages belonging to the same language family, and Vietnamese is a language with an isolating morphology, this judgement seems intuitive.

If cross-lingual transfer across stacked language adapters is facilitated by similarity between languages in terms of morphological complexity, the stack with Finnish should perform higher than the one with Vietnamese. However, the graph in 4.5 shows that



(a) **low-dttr** experiment



(b) **high-dttr** experiment

Figure 4.4: Diagrams representing the flow of experiments involving adapter stacks. In either experiments, another language adapter is stacked on top of the relevant language adapter for Hungarian. In the case of **low-dttr**, this is the Finnish adapter (see Figure 4.4a), while in the case of **high-dttr**, this is the Vietnamese adapter (see Figure 4.4b). Resulting sentence representations are then used in the cross-lingual sentence retrieval task to identify translation pairs in using cosine similarity.

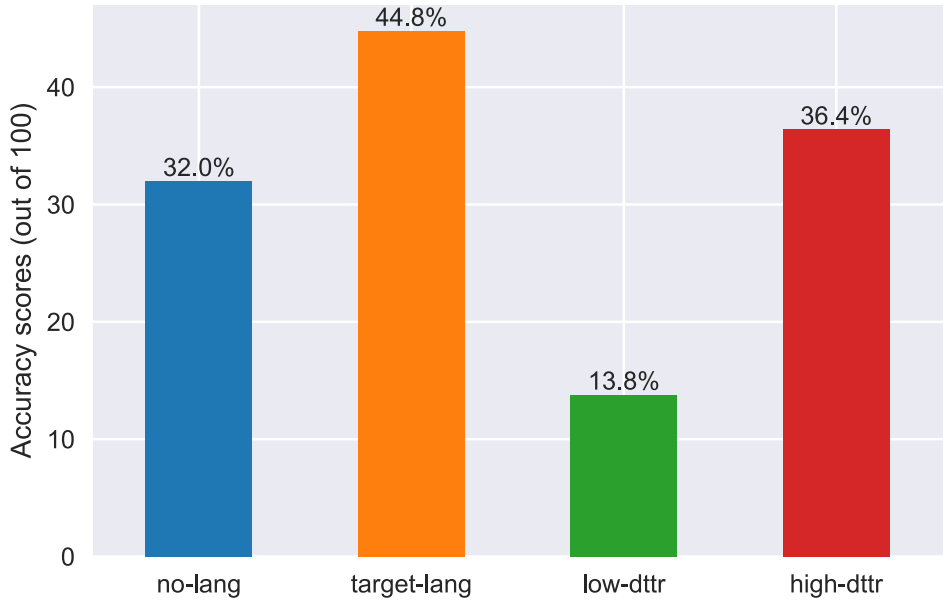


Figure 4.5: Accuracy scores on the cross-lingual sentence retrieval task between Hungarian and English sentences comparing the two baselines, **no-lang** and **target-lang**, and two experiments involving adapter stacks, **low-dttr** (stacked Finnish adapter) and mbox**high-dttr** (stacked Vietnamese adapter). Best performance is achieved using the **target-lang** baseline, while the worst performance is yielded by the **low-dttr** experiment.

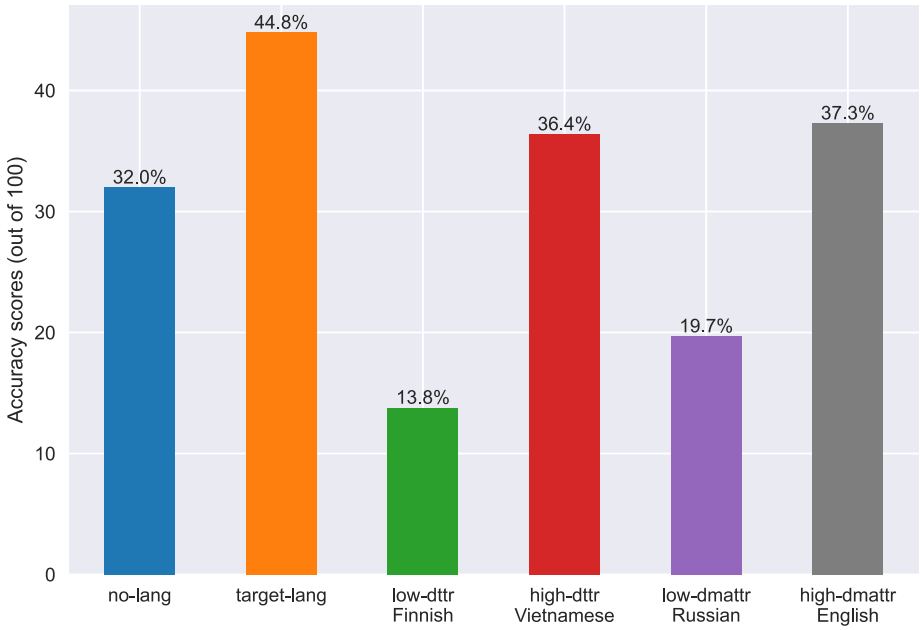


Figure 4.6: Accuracy scores for all the experiments involving Hungarian. The best result is yielded when using the **target-lang** baseline. Despite my hypotheses, adapter stacks involving Vietnamese and English, languages that have high distance scores from Hungarian in terms of morphological similarity outperform the adapter stacks that pair Hungarian with more similar languages in terms of morphological complexity.

despite this, performance is considerably higher when the **high-dttr** setup is used than when using **low-dttr**. In fact, **low-dttr** harms performance even compared to **target-lang**.

Adapter stacks based on moving average type-token ratio

The other corpus-based measure I employ is the moving average type-token ratio, averaged over a moving window of 500 words. As I described in Section 3.2.2, this measure is meant to be more robust to effects of text length. This is something that affects even parallel corpora. The length of the New Testament for different languages, for instance, is not equal, as shown in Table 4.1. It is possible then that this measure would result in a more accurate picture regarding the relative similarity in terms of morphological complexity between different languages.

Otherwise, the formula for determining the similarity between languages is the same for the moving average type-token ratio as for the overall type-token ratio. Similarly to distance measured in terms of overall type-token ratio, low distance in terms of moving average type-token ratio is also expected to have a positive influence on the performance in thecross-lingual sentence retrieval task task, while high distance is predicted to have a comparatively negative impact on performance.

The language with the lowest distance score to Hungarian in terms of moving average type-token ratio is Russian (**low-dmattr** setup), whereas the language with the

highest such distance score is English (**high-dmattr** setup). Expectations are similar to using overall type-token ratio: the inclusion of Russian should contribute to more successful cross-lingual alignment. However, since alignment has to be achieved with English sentences, it can be expected that the combination with the English language adapter outperforms not only the **low-dmattr** setup, but also the next highest adapter combination, with Vietnamese (**low-dttr**) (see Figure 4.6).

In fact, it appears that using English in the **high-dmattr** contributes to higher performance than Russian. The use of English even yields better accuracy scores than all other adapter stacks. This phenomenon will be addressed in the next chapter where the full results of all the eight test languages are described.

4.4 Summary

In this chapter, I presented the preparatory work that was required to operationalise my research questions, also outlining the experiments that I carried out. Due to the large number of experiments and languages, I used the example of one language, Hungarian, to make the experimental setups more legible. In Section 4.1, I started by briefly reiterating my research questions first introduced in Chapter 1.

In Section 4.2.1, I described the process of quantifying morphological complexity of the different languages using the corpus-driven measures of type-token ratio and moving average type-token ratio. I also expanded on how I processed the Parallel Bible Corpus to yield parallel corpora for the different languages. Additionally, I made observations about the idiosyncrasies of establishing distances between different languages using these morphological complexity measures.

In the remainder of Section 4.2, I outlined how I prepared the evaluation sets on the cross-lingual sentence retrieval task for each language. Then I described how I arrived at the final sample of languages that I used for my experiments, also providing the characteristics of the individual languages.

In the last section, Section 4.3, I describe the individual experimental setups that I did for each of the eight languages in the final sample using Hungarian as a working example. These included two baseline experiments, **no-lang** and **target-lang**, and four adapter stacks: **low-dttr** and **high-dttr**, and **low-dmattr** and **high-dmattr**.

Chapter 5

Results

In this chapter, I describe the results on cross-lingual sentence retrieval task of the experiments introduced in Chapter 4. As mentioned in Chapter 3, the task can be interpreted as a ranking task, and as such, I carry out evaluation by assessing either if the correct translation has the most similar sentence-level representation to the target sentence ($k = 1$), or if it within the first three most similar ones ($k = 3$). The experimental setups for each language consist of the baseline experiments, **no-lang** and **target-lang**, as well as four other experiments involving language adapter stacks. Language adapters are paired on the basis of distance from other languages as measured by morphological complexity scores, overall type-token ratio (**low-dttr** and **high-dttr**), and moving-average type-token ratio (**low-dmattr** and **high-dmattr**).

There are many experiments I implemented. To make the results more legible, I focus discussion on the most and least successful experimental setups, and on common trends that can be observed across the languages. Since the test sets on the cross-lingual sentence retrieval task are not parallel, I compare results across languages by using improvements measured in percentages compared to the two baseline experiments.

5.1 Best and worst results

Cross-lingual sentence retrieval across 1,000 sentence pairs is far from a trivial task. This is shown by the low accuracy scores the random baseline achieves on it (0.4, see Section 4.3.1 for details). It is also demonstrated by the fact that the best result achieved on it

Language	Best setup	Improvement
Amharic	**low-dttr** (Arabic)	+16.2%
Icelandic	**high-dttr** (Vietnamese)	+4.2%
Meadow Mari	**high-dmattr** (English)	+25.3%
Northern Sami	**high-dmattr** (English)	+16.8%

Table 5.1: The best performing experimental setups for low resource languages in my sample when the accuracy is calculated based on $k = 1$. Languages with similar setups are grouped together. Improvement refers to the improvement achieved over the **target-lang** baseline measured in percentages. Where the best performing experimental setups involve adapter stacking, the adapter stacked on top of the target language adapter is in parenthesis after the setup name.

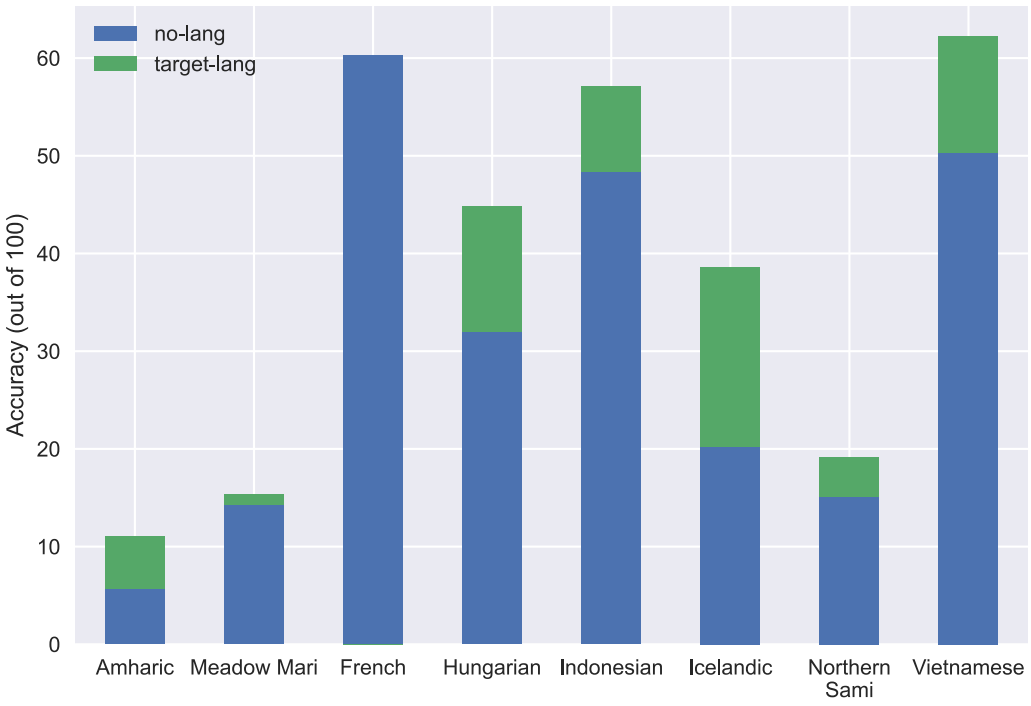


Figure 5.1: The difference between the accuracy scores achieved on the cross-lingual sentence retrieval task when using the **no-lang** baseline and **target-lang** baseline. In the case of French, the target language adapter results in a performance decrease of 13.77%, hence its value cannot be represented. The results are presented across languages for convenience and should not be directly compared due to the differences in test data content.

across languages and experiments is an accuracy score of 60.3 when using the original language model representations to connect sentence pairs across English and French. If strong alignment with English correlates with strong language model knowledge, as asserted in Section 4.3, this indicates that multilingual BERT knowledge still requires improvement even for high resource languages, such as French.

The graph in Figure 5.1 delivers two main messages. First, that the original language model knowledge is considerably lower on all other languages than on French. Second, the **target-lang** baseline improves performance on all languages aside from French. This is done to a varying degree. The highest contribution is on Amharic and Icelandic, 95% and 91% increase, respectively, compared to the **no-lang** baseline. On other languages, the improvement is more modest. The **target-lang** setup results in a +40% performance benefit for Hungarian, +24-26% for Northern Sami and Vietnamese, +18% for Indonesian, and a mere +8% for Meadow Mari.

For languages that can be considered at least moderately high resource ones, such as French, Hungarian, Indonesian, and Vietnamese, the setups involving stacked language adapters do not contribute to any performance increase compared to the **target-lang** baseline. This indicates that for these languages, multilingual BERT knowledge cannot be improved by stacking language adapters.

On the other hand, for low resource and zero-shot languages, like Amharic, Icelandic, Meadow Mari and Northern Sami, the highest accuracy scores are reached when adapter stacks are used. Table 5.1 shows which experiments and which adapter combinations contribute to the highest performance for each of these languages, with improvement shown over the **target-lang** baseline.

According to my hypotheses, improvement is expected to be highest when the languages involved in the adapter stack are most similar to each other in terms of morphological complexity due to a boost in cross-lingual transfer. Despite this, it is only in the case of Amharic where the highest score is achieved with the **low-dttr** experiment. For the other languages, performance on the task is most improved when stacking languages that are not similar in terms of morphological complexity. This is surprising as based on prior research I expected that cross-lingual transfer would be hurt across languages that are typologically diverse.

It is especially intriguing that multilingual BERT performance increases on Meadow Mari and Northern Sami the most when their adapters are combined with the English language adapters. Stacking the English language adapter also performs comparatively well for Hungarian (see Section 4.3.2). As essentially what the cross-lingual sentence retrieval task directly measures is alignment with English, it is also possible that adapter stacks with English will always be more successful than what can be expected based on pure typological similarity. However, the graph in Figure 5.2 shows that the fact that the English language adapter contributes to a better alignment with the English sentences is clearly not the only factor at play here. While stacking the English language adapter on top of the target language adapter contributes to a higher performance than the **no-lang** baseline on all test languages except for French, it only results in superior performance compared to **target-lang** only for Meadow Mari and Northern Sami.

Table 5.2 shows which experimental setups performed lowest for each language. The findings partially support the hypothesis that cross-lingual transfer is hurt by dissimilarity between the languages in terms of morphological complexity. For four out of eight languages, the worst accuracy score is yielded when the Korean language adapter is stacked on top of the target language adapter in the **high-dmattr** setup.

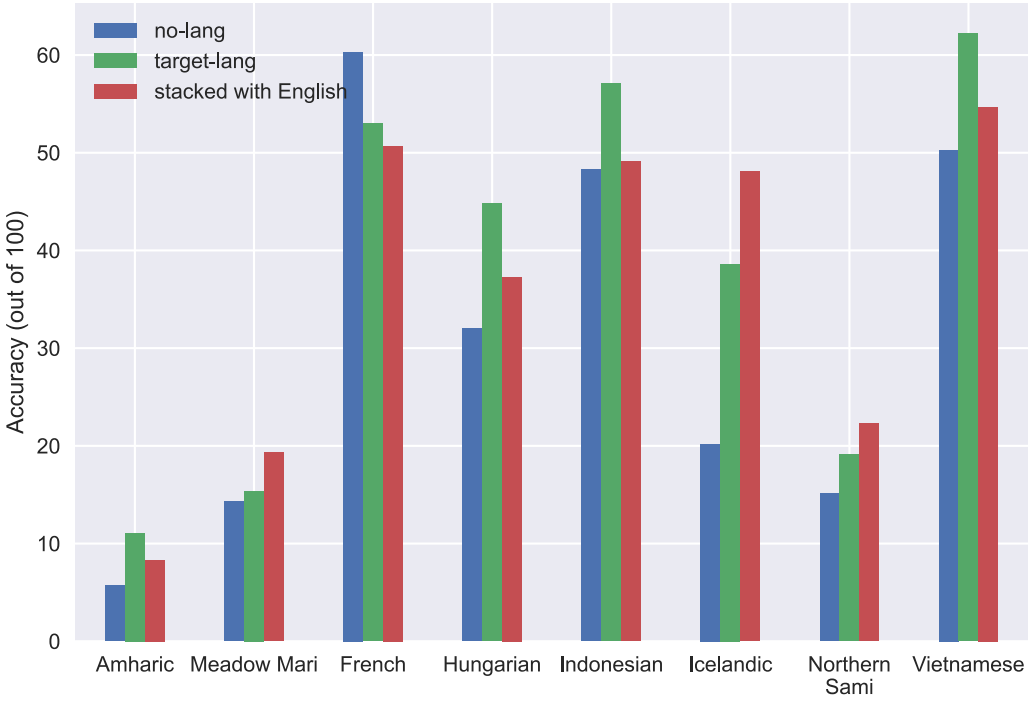


Figure 5.2: Accuracy scores on the cross-lingual sentence retrieval task comparing the **no-lang** and **target-lang** baselines and when the English adapter is stacked on top of the target language adapter. The results are presented across languages for convenience and should not be directly compared due to the differences in test data content.

Language	Worst setup ($k = 1$)	Decrease
Amharic	**no-lang**	−48.7%
French	**high-mattr** (Korean)	−76.8%
Indonesian	**high-mattr** (Korean)	−61.5%
Icelandic	**high-dmattr** (Korean)	−56.7%
Vietnamese	**high-mattr** (Korean)	−57.7%
Meadow Mari	**low-dmattr** (Erzya)	−43.5%
Northern Sami	**low-dttr** (Meadow Mari)	−23.6%
Hungarian	**low-ttr** (Finnish)	−69.2%

Table 5.2: Experiments yielding the worst accuracy scores for each language in the sample. Languages with similar setups are grouped together. Decrease refers to the decrease in score compared the **target-lang** baseline measured in percentages. Where the worst performing experimental setups involve adapter stacking, the adapter stacked on top of the target language adapter is in parenthesis after the setup name.

Korean is an agglutinative language, and in this it differs in its morphology from the other languages it is paired with. This might contribute to this particular combination yielding the worst results.

In another set of cases, worst performance is reached when the language adapter of a zero-shot language is stacked on top of the target language adapter. This is the case for both Meadow Mari and Northern Sami, both stacked with language adapters of other Uralic languages. It is possible that the language adapter of a zero-shot language is comparatively so undertrained that it actively hurts language model knowledge. This finding is further corroborated by the fact that all experiments stacking a zero-shot language adapter on top of a target language adapter results in a low performance.[1]

Amharic is an exception in that the worst performance on it is achieved in the **no-lang** baseline. This is likely because Amharic is written with the Ge'ez script that the multilingual BERT model did not encounter during its pretraining. It is possible that the model knowledge on Amharic is so low that any added adapters can contribute to a comparative improvement in this knowledge.

Finally, in the case of Hungarian, a marked decline can be observed when the Finnish language adapter is stacked on top of the target language adapter as it can also be seen in Section 4.3.2. This is a somewhat perplexing result, considering Finnish is a language that is closest to Hungarian in terms of type-token ratio, they both belong to the same Uralic language family, and that Finnish is not even remotely a zero-shot language.

5.2 Analysis of low resource languages

Since I conducted a large number of experiments, it is necessary to focus the discussion by restricting the number of languages to consider. In the following, I decided to address the results achieved on low resource languages when using the different adapter stacks. These are Icelandic, and the three zero-shot languages, Amharic, Meadow Mari, and Northern Sami.

The graph in Figure 5.3 shows the relative impact of language adapter stacks expressed in terms of percentages for each of these four languages compared to the **target-lang** baseline. The graph shows that language adapter stacks including languages that are similar to each other in terms of morphological complexity, i.e., the **low-dttr** and **low-dmattr** experimental setups, either do not reliably enhance, or in fact explicitly harm, the performance of multilingual BERT on the cross-lingual sentence retrieval task, except for Amharic. This contradicts the hypothesis also phrased in the previous section.

As also mentioned in the previous section, experiments involving the stacked adapters of zero-shot languages seem to harm language model knowledge on the target language, contributing to a lower performance on the evaluation task. These involve **low-dmattr** for Meadow Mari, paired with the Erzya language adapter, **low-dttr** for Icelandic, where the Ilocano language adapter is stacked, and the stacking of the Meadow Mari adapter for Northern Sami (**low-dttr**).

There are a couple of other languages that even though are not zero-shot, contribute to an overall poor performance when their language adapter is stacked, similarly to how the stacked Finnish adapter hurt performance with Hungarian. Estonian is such an example when paired with Meadow Mari, even though they are both Uralic languages.

[1] See Appendix A for full results.

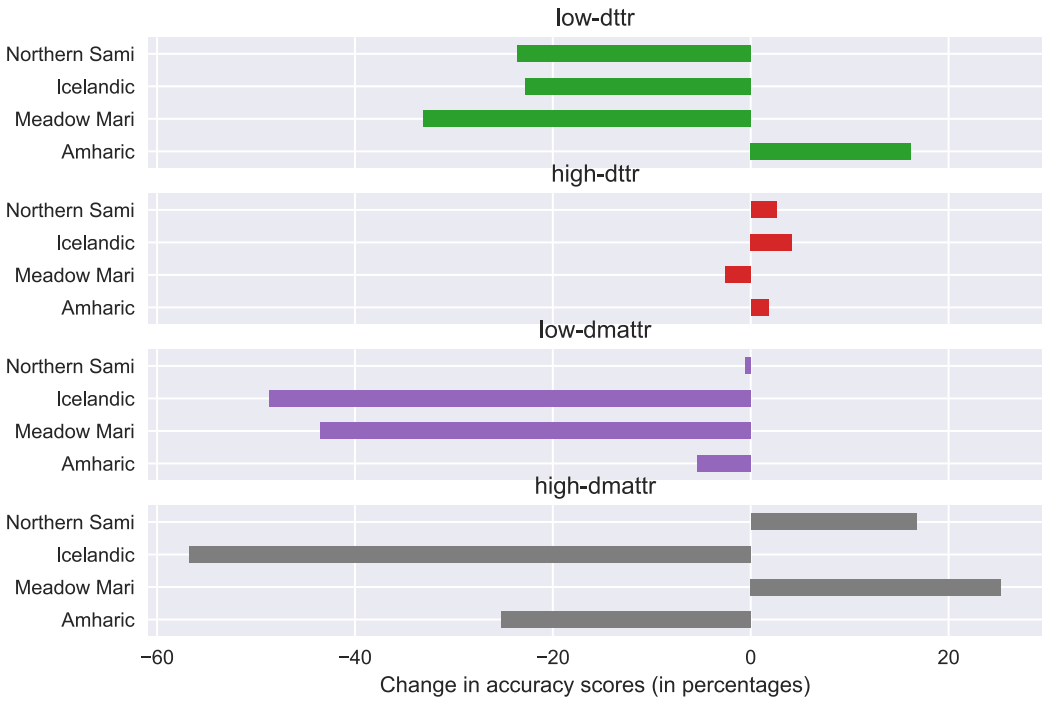


Figure 5.3: Relative change in percentages in performance compared to the **target-lang** baseline for the four low resource languages in my sample.

Additionally, Greek seems to hurt performance on Icelandic more than the zero-shot Ilocano. This is surprising in light of the fact that Greek is paired with Icelandic in the **low-dmattr** experiment, and that it has significantly more training data available than Ilocano. Despite this, the setup with Ilocano results in a performance decrease of 22.8% compared to the much more significant decrease with Greek, of 48.7%.

Finally, variability in the results shown on the graph in Figure 5.3 is much smaller for the **high-dttr** setups than for other experiments. In all cases, it is Vietnamese whose language adapter is stacked on top of the target language adapters. Despite the hypotheses, it appears that Vietnamese may contribute to model knowledge on low resource languages, resulting in a slight, 2.6% drop in performance only for Meadow Mari.

Chapter 6

Discussion

In this chapter, I set out to do discuss the results introduced in Chapter 5. In Section 6.1, I discuss the main trends that can be observed across languages, with special focus on relating findings to my hypotheses, while also highlighting questions that I could not answer. The rest of the chapter focuses on the set of low resource languages in my sample, Icelandic, and the zero-shot Amharic, Meadow Mari, and Northern Sami. In Section 6.2, I show my error analysis, focusing on sentence types affected by different experiments. I also analyse the influence of factors related to characteristics of source sentences, including length and tokenizer coverage. Finally, in Section 6.3, I provide recommendations for future work.

6.1 Addressing research questions

The results in Chapter 5 showed us how much adapters of different languages, or different experimental setups, contribute to performance on cross-lingual sentence retrieval across languages. The aim of the experiments was to improve alignment between sentence-level representations that the multilingual BERT model creates for English and various target languages. While this is an indirect way to assess language model knowledge on specific languages, I believe that this is a valid trade-off in exchange of the higher coverage of the task. Also, as mentioned in Chapter 4.3, the degree of this alignment shows the quality of the model knowledge on the target language, as language model representations can be expected to be the most robust for English. By increasing alignment between representations between English and another language, model knowledge should improve on the other language (Jones et al., 2021).

In the following, I will address my research questions, explicitly comparing my prior expectations with my findings. In this, I do not directly compare accuracy scores since the evaluations sets are not completely parallel across languages (5).

Can the stacking of language adapters contribute to increased model performance on downstream tasks?

My first research question pertains to the success of the methodology I use in this thesis. If the answer to this question is negative, that indicates that stacking language adapters results in the obscuring of crucial language-specific information in all cases, and that cross-lingual transfer is better induced and investigated with an alternative methodology.

Language	Best combination	Improvement
Amharic	Arabic	+16.2%
Icelandic	English	+24.6%
Meadow Mari	English	+25.3%
Northern Sami	English	+16.8%

Table 6.1: The best improvements achieved on low resource languages using an adapter stack in percentages. The Best combination column shows which language adapter was stacked on top of the target language adapter to yield the improvement. Improvement is in percentages, meant over the **target-lang** baseline.

My original hypothesis was that stacking language adapters can in fact increase model performance. I stated that a successful combination of the information encoded inside the two language adapters is in fact possible through cross-lingual transfer. My results seem to corroborate this hypothesis. While not every language adapter stack contributed to a higher accuracy score in the cross-lingual sentence retrieval task, certain adapter stacks did improve performance. This entails there is a certain additive nature to the information stored inside language adapters, and thus they can be used to investigate factors interacting with cross-lingual transfer.

Which languages benefit the most from stacking language adapters?

It has been mentioned in this thesis that language model performance is not equal across all languages: see for instance Section 2.3. This performance depends on various factors, including the size of the pretraining data for the language that was used to create the language model.

When it comes to improving cross-lingual transfer using stacked language adapters, I expected more success for languages that are low resource and zero-shot. My reasoning is related to the amount of training data available. Language adapters are pretrained on this data, so if less of it is available, it makes it more likely that the language adapters are undertrained. This way the language adapter can benefit from the additional information provided by the second language adapter.

The experimental results seem to support my prior assumptions. The only languages out of my sample where an improvement – however little – is achieved when using stacked language adapters over the baseline (**target-lang**) are the low resource languages, Icelandic, Amharic, Meadow Mari, and Northern Sami. For moderately high and high resource languages such as Hungarian, Indonesian, Vietnamese, and French, best performance is achieved either when only the target language adapter is used, or, in the case of a truly highly resource language such as French, when the multilingual BERT model is used without any adapters.

While it appears that within the group of low resource languages the performance improvements achieved on zero-shot languages, Amharic, Meadow Mari, and Northern Sami, are higher than on Icelandic, this is shown to be an accidental factor when combining Icelandic with the English language adapter. This was done not as part of the original experiments, but to assess the relative contribution of the English language adapter (see Figure 5.2). On Icelandic, the use of English yields an improvement of +24.6%, surpassing almost all other improvements (see Table 6.1).

Language	English	**French**	German	Spanish
Word count	4.1B	**1.5B**	1.4B	1B
Language	Russian	Arabic	**Vietnamese**	**Hungarian**
Word count	919M	365M	**304M**	**200M**
Language	**Indonesian**	Finnish	Korean	Greek
Word count	**172M**	149M	146M	125M
Language	Armenian	Estonian	Burmese	**Icelandic**
Word count	95.1M	52.7M	19.8M	**12.1M**
Language	Ilocano	**Meadow Mari**	**Amharic**	Erzya
Word count	3.3M	**2.6M**	**1.7M**	1.5M
Language	**Northern Sami**	Wolof		
Word count	**690K**	658K		

Table 6.2: Word counts of different Wikipedias in a selection of languages which are either in the test sample or whose language adapters are used in adapter stacks. Languages in bold letters are in the test sample. Data retrieved from `https://meta.wikimedia.org/wiki/List_of_Wikipedias/Table`.

Which factors determine which language adapters are useful to combine?

Besides the size of the available training data, language model performance on a specific language depends on other properties relating to vocabulary, orthography, and typological features (see Section 2.3 for more details).

As discussed in Section 2.3.3, prior research has shown that typological similarity between languages may improve cross-lingual transfer. I built on this when I hypothesised that cross-lingual transfer that stacked language adapters induce would be the strongest between languages with a high degree of typological similarity between them. I approximated this typological similarity as the similarity in morphological complexity between languages. I expected that when a language is close to the target language in terms of morphological complexity, stacking its adapter on the target language adapter would result in an improvement in downstream performance on the task of cross-lingual sentence retrieval. This would reflect that model knowledge increases on the target language.

In practice, this hypothesis was not corroborated by my results. This means there must be some other factors at play on deciding how stacked language adapters might help or hinder downstream task performance. In the following, I examine linguistic features that might contribute to cross-lingual transfer between stacked language adapters.

There is only one example where it appears that a language adapter of a language with comparable morphological complexity to the target language improves performance on cross-lingual sentence retrieval. This example is when the Arabic language adapter is stacked on top of Amharic, a zero-shot language. But even for this combination, it has to be noted that even then the two languages belong to the same language family. This makes it possible that there are other factors at play in this cross-lingual transfer.

Two languages belonging to the same family might share various features between each other that might positively influence cross-lingual transfer. These features might include morphology, of course, but also a similar word order or shared vocabulary items. For the sake of discussion, let's examine these features between Arabic and Amharic.

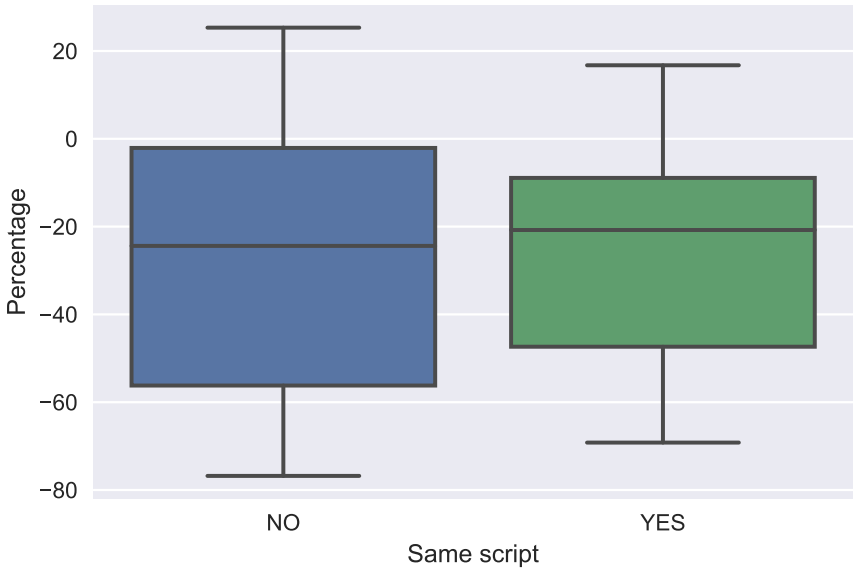


Figure 6.1: The contributions of different adapter setups in percentages based on whether the languages of the two adapters stacked share a script. The improvements fall in the same range, showing that shared script does not predict if a particular adapter combination would be useful.

Word order seems to broadly differ between Amharic (SOV, subject, object, verb order), and Modern Standard Arabic (VSO, verb, subject, object) language (Dryer and Haspelmath, 2013). Other Arabic dialects might also have an SVO word order, but this still does not match with Amharic, indicating that in this scenario, this is not the factor that contributes to cross-lingual transfer between the two language adapters.

Another possibility is lexical overlap, but Arabic and Amharic do not share scripts. This would likely obscure any potential connections in terms of vocabulary between them. Additionally, since the Ge'ez script Amharic is written with is not seen by the multilingual BERT model during its pretraining, most Amharic vocabulary items are in fact not known by the original model. In practice, this means that most Amharic subword tokens are replaced by the UNK (*unknown*) token, accounting for the low performance of the model without adapters. This also makes it unlikely the model recognises lexical overlaps between the two languages. There might be other factors at play between these two languages, but this needs further investigation.

It is worth investigating whether lexical overlap would contribute to a more successful cross-lingual transfer between language adapters across the board. This is unlikely to be high if the two languages do not share a script. This might also explain why stacking Korean on top of other language adapters impacts performance to such a negative extent. The boxplot in Figure 6.1 tracks how the distribution of percentages might depend on whether the two languages whose adapters are stacked share a script or not. It is clear that the contribution of adapters do not depend on this factor alone.

When it comes to other languages, cross-lingual transfer between stacked language adapters does not seem to depend on the morphological similarity between the lan-

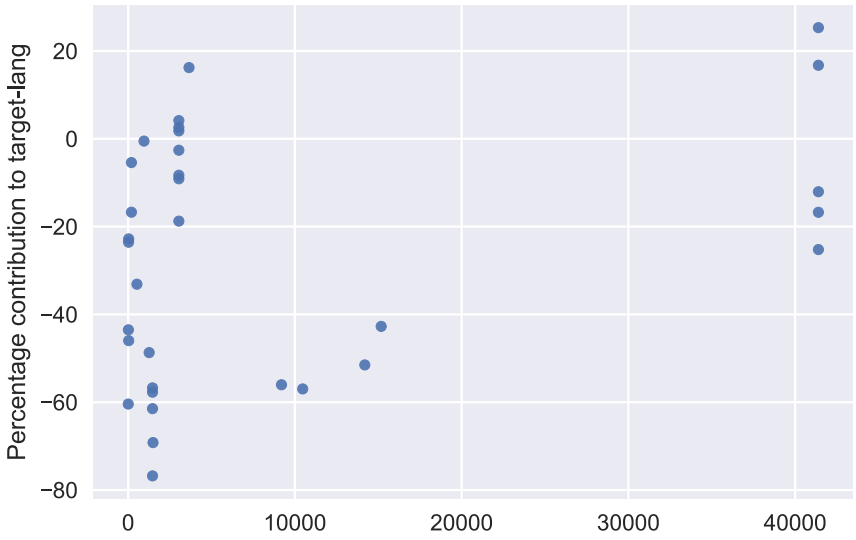


Figure 6.2: Percentage contributions over the **target-lang** baseline of different stacked language adapters plotted against the pretraining data size (shown in units of 100K.

guages involved. For most languages, the most significant contribution – or the least negative impact – is provided by the English and the Vietnamese language adapters, typically involved in the **high-dmattr** and **high-dttr** experiments, respectively.

English has a positive impact as a stacked adapter on top of Meadow Mari and Northern Sam adapters. It can be hypothesised that this is because the English adapter contributes to better aligning representations with the English candidate translations encoded with the English adapter. However, this explanation is not consistent with other results and the graph in Figure 5.2. When combined with English, the Amharic, Hungarian and Vietnamese adapters achieve poorer results than on the **target-lang** baseline. It is possible then that there is more English content, e.g. technical vocabulary, in the Meadow Mari and Northern Sami texts that help alignment when the English language adapter is stacked.

Besides English, Vietnamese is the other language whose adapter contributes relatively positively all across the languages when stacked on the target language adapter. Training data size could be one explanation for the consistently good cross-lingual transfer provided by language adapters of these two languages. The more training data there is, the more language-specific knowledge the adapters should be able to acquire. This is in line with my other hypothesis, that adapters trained on more training data should be able to contribute to cross-lingual transfer better than adapters that are trained on less training data.

Table 6.2 shows the Wikipedia sizes of the different languages which are in the test sample or whose adapters were used in different configurations. This data was used to pretrain the language adapters on AdapterHub (Pfeiffer et al., 2020a). It demonstrates that English and Vietnamese are among the top languages in terms of training data size available. This would make their adapters better trained than most other language adapters in the sample. But, as mentioned above, stacking the English language adapter

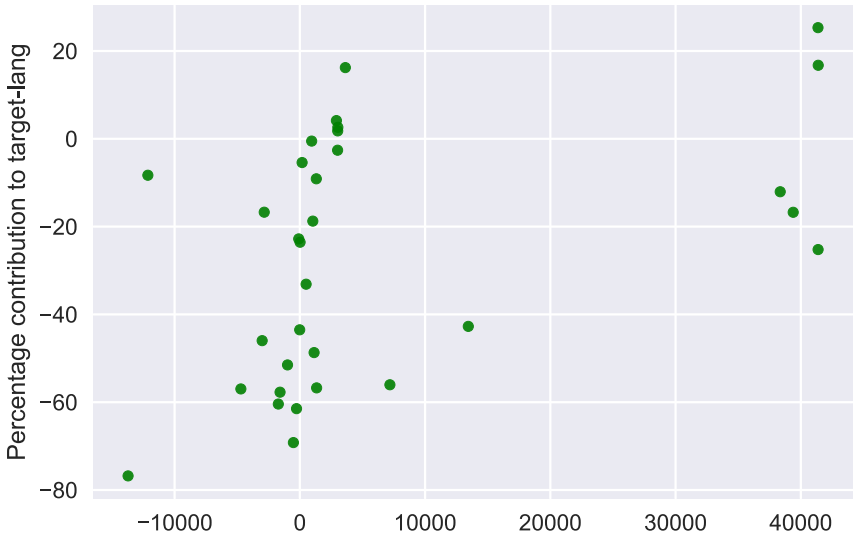


Figure 6.3: Percentage contributions over the **target-lang** baseline of different stacked language adapters plotted against the difference in pretraining data size between the target and stacked language adapters (shown in units of 100K).

on top of the target language adapter does not result in a consistently good performance on all languages.

Additionally, the size of the Vietnamese training data is surpassed by some truly high resource languages such as Russian (919 million words), German (1.4 billion words), French (1.5 billion words), and, of course, English (4.1 billion words). Still, even where Vietnamese does not actively help in performance, its adapter consistently yields less of a performance hit than these other languages, save for English.

The graph in Figure 6.2 further shows that the contribution of different stacked language adapters depends on more than the training data size of the adapters alone. Language adapters with training data sizes of around and above 1 billion words perform worse than many adapters trained with much less data. Additionally, while the highest contribution is by English, the language adapter with the largest amount of training data available, its contribution when combined with certain languages is lower than many language adapters trained on much lower amounts of data.

Another potential explanation might be that it is not the amount of training data size available, but the difference between the training data sizes of the two language adapters that might predict whether a certain adapter combinations induce cross-lingual transfer. According to this hypothesis, the highest performance can be reached when the target language adapter is undertrained with low amounts of training data, and the language adapter stacked on top of it is trained on much more data. The graph in Figure 6.3 shows that this is not consistent with the results. In itself, the difference between language adapters in terms of training data does not seem to predict whether a certain adapter combination will be successful. Performance is not guaranteed to improve regardless of whether the difference between the pretraining data of the languages is small or large, negative (for the target language adapter), or positive.

After investigating a number of linguistic factors, such as similarity in morphological complexity, and extra-linguistic factors such as training data sizes and the actual scripts a language is written with, I could not identify a clear relationship between these factors and which language adapter combinations help cross-lingual transfer, increasing performance for the cross-lingual sentence retrieval task.

The comparatively positive role of the Vietnamese adapter when combined with target language adapters is perplexing and may lie in the unique properties of the language. Vietnamese certainly has features that make it stand out compared to other languages. First, its character inventory is large due to the inclusion of various tonal marks in its alphabet. Additionally, Vietnamese has a unique orthographic convention of using the whitespace character not only as a word separator, but also as a separator of syllables. See the example in (12) from Nguyen and Tuan Nguyen (2020), where the word *nghiên cứu viên* 'researcher' is in fact made up of three syllables separated by whitespace characters just like the other words in the sentence.

(12) Tôi là một nghiên cứu viên
I am a researcher – –
'I am a researcher'

Perhaps the large character inventory allows a relatively good lexical overlap with other languages, which is conducive to cross-lingual transfer. However, it is not clear how or why the quasi-syllabic nature of the Vietnamese texts would relate to its cross-lingual transfer capabilities on other languages.

While language adapters of English and Vietnamese provide consistently good cross-lingual transfer, certain other languages seems to hurt model performance on target languages. Some of these are perplexing, for instance why the Finnish and Estonian language adapters would contribute to low performance with Hungarian and Meadow Mari (both in the **low-dttr** experiment), respectively. Further investigation is needed to find an explanation for these findings.

6.2 Error analysis

In this section, I focus on observing the interaction of the sentence types and the different experiments in the cross-lingual sentence retrieval task, with special focus on the error cases. After a qualitative analysis of error types, I also quantitatively analyse the results based on a few different factors, including source sentence length and the ratio of UNK tokens.

6.2.1 Sentence types

In this section, I discuss common sentence types in the cross-lingual sentence retrieval task with special focus on what cues the multilingual BERT model might utilise when creating sentence-level representations. In this discussion I focus on the low resource languages in my sample, Icelandic, and Amharic, Meadow Mari and Northern Sami. This is both to limit discussion, and because my results show that these are the languages that most benefit from cross-lingual transfer induced using stacked language adapters. To highlight the contrast between using the target language adapter and a

stacked language adapter setup, I focus on the **target-lang** baseline and the adapter combination that reaches the highest performance.

Amharic is paired with Arabic in the **low-dttr** experiment. Both for Meadow Mari and Northern Sami, the best performing adapter combination is with English, in the **high-dmattr** setup. Finally, for Icelandic, I made the decision to use the **high-dttr** setup with the Vietnamese language adapter, the only case where the performance impact is negative compared to the **target-lang** baseline. While the overall highest performance is reached when the English language adapter is used, it was not part of the original set of experiments.

The main pattern that can be observed is that sentence-level representations of low resource languages created using multilingual BERT seem to be especially sensitive to certain elements. These consist of special characters (such as punctuation), Latin characters in a language with a non-Latin script, numerical characters, and word-level semantics (overlapping meaning). It also appears that a stacked language adapter might accentuate this sensitivity, especially when it comes to the last three categories.

In the following, I discuss specific examples that illustrate these patterns. In the tables that follow, I use a colour scheme. If the cell with the **True match** column is red, that means that neither the **target-lang**, nor the best performing adapter stack could rank the correct translation in the first three candidates. If the colour is yellow, the correct translation is returned by at least one of these experiments within the first three results. Finally, if it is green, then the true match is ranked first by at least one of these setups.

Special characters

In a subset of the cases, it appears that the sentence representations the multilingual BERT model creates are influenced to a large degree by the special characters they contain. Example for these special characters are square brackets ([]), colon (:), exclamation mark (!), and the percentage symbol (%). It is not predictable whether the **target-lang** baseline or a specific adapter stack would perform better in matching segments that contain these special characters. The presence or absence of punctuation can help the model in connecting two sentences that might be translations of each other. In other cases, this might be the cause of erroneously connecting sentences.

For examples, see the table in Figure 6.4. It can be seen that in many cases, the matching special character makes it likely for the model to rank high other sentences that also contain the same character.

Latin characters

Multilingual BERT representations are also sensitive to the presence of Latin characters in texts that are otherwise written in a language that does not use the Latin alphabet. This is especially true when the proportion of subword tokens that contain Latin characters is high in the given sentence. This makes it more likely that the overlapping subwords would increase similarities between the two sentence representations.

The table in Figure 6.5 features a few examples from Amharic (written with the Ge'ez script) and Meadow Mari (written with the Cyrillic alphabet) that illustrate this explanation. For instance, in example 1 with Amharic, the abbreviation written with Latin letters, *AUD/CUP*, is broken down into five subwords (see in (13)).

(13) AU, ##D, /, C, ##UP

Index	Language	Source text	True match	Predicted match	Correct setup
1	Amharic	[የግርጌ ማስታወሻ]	[Footnotes]	[Credit Line]	Neither
2	Meadow Mari	Жап эртен:	Time Elapsed:	Channels:	k=1 high-dmattr
3	Meadow Mari	Шочмо кечет дене!	Happy birthday to you!	Good luck!	k=3 high-dmattr
4	Meadow Mari	%s йодеш %s гынат, тудым ястарен кертме огыл.	%s requires %s, but it failed to unload.	Unable to buzz, because %s does not support it or does not wish to receive buzzes now.	k=1 high-dmattr
5	Northern Sami	Ollu giitu!	Thank you very much!	You are welcome! *or:* No worries!	Both k=3
6	Northern Sami	Válddahus:% 1	Descriptio n: %1	View Attachment : %1 *or:* Group: %1	Neither

Figure 6.4: Examples in which the sentence representations the language model generates are influenced by special characters such as punctuation.

The total subword count in this sentence is 7, thus the proportion of subwords written with the Latin script is high, influencing the language model representation. On the other hand, in examples 2 and 4, the Latin characters only generate two subwords (out of eight for Amharic) and five subwords (out of twenty for Meadow Mari), respectively (see in (14)). This means the total proportion of subwords is low, only $\frac{1}{4}$ for both.

(14) a. guest, ##room
b. ", i, ##cq, ", URL

In other cases where the proportion of Latin to non-Latin subword tokens is not particularly high, such as in example 5 in Figure 6.5, the language model might or might not have enough information to create well-aligning sentence representations between English and the target language. However, adding English in the **high-dmattr** setup seems to contribute to increased model sensitivity to overlapping tokens.

Numbers

Besides punctuation and Latin characters, the presence of numbers might also affect the sentence representations that multilingual BERT generates. When looking at the table in Figure 6.6, we can observe cases where matching numbers contributes to the success of cross-lingual sentence retrieval. This is especially likely for example 2, but also probable for example 4. In these sentences, the additional language adapter does not seem to make a difference as both experiments are successful.

In a subset of cases, such as in example 1, the model representation is not sensitive to the number when only the target language adapter is used. It appears that the

Index	Language	Source text	True match	Predicted match	Correct setup
1	Amharic	AUD/CUP ዋጋ ዝርዝሮች	AUD/CUP rate details	AUD/CUP rate details	Both
2	Amharic	እርሱ guestroom አቅጣጫ ደስተኛ እርምጃ ወሰደ.	He took an unhappy step toward the guestroom.	He went to the student's house. *or:* Tom always drinks coffee in the morning.	Neither
3	Meadow Mari	delbuddy' команде ыш шуктэлт	delbuddy' command failed	delbuddy' command failed	Both
4	Meadow Mari	Палемдыме команде "icq" URL-ым келыштарышаш мо	Whether the specified command should handle "icq" URLs	Novell GroupWise Messenger Protocol Plug-in	Neither
5	Meadow Mari	libpurple дене колтымо шифр-влакым терга.	Tests the ciphers that ship with libpurple.	The SIP/SIMPLE Protocol Plugin	k=1 high-dmattr
6	Meadow Mari	GIMP сӱретым тӧрлатыше	GIMP Image Editor	Bluetooth settings…	k=1 target-lang

Figure 6.5: Examples in which the sentence representations the language model generates are influenced by Latin characters in the case of languages which are not written in the Latin alphabet.

Index	Language	Source text	True match	Predicted match	Correct setup
1	Amharic	በ ኤሪን ኢሞን ጋቪንኤፕሪል 2015	By Erin Imon Gavin April 2015	As it can be recalled, at the time the Muslim community wanted to reorganize the 'Islamic Affairs Council' anew. *or:* Happy new year 2015 – What's in store?	Neither
2	Amharic	1.1.1 አስተያየቶች	1.1.1 Comments	1.1.1 Comments	Both
3	Icelandic	að þeim sem gefa blóð hefði fækkað um 25 af hundraði í júlí 1983.	that blood donations went down 25 percent during the month of July 1983 .	The 50,000 coconut trees and 40 new homes that were part of the three - million - dollar rehabilitation plan were abandoned . *or:* The divorce rate is rising sharply and venereal diseases are epidemic .	k=3 high-dttr
4	Northern Sami	EBU nannii skábmamánu 17. b. 2017, ahte 43 riikka oassálastá jagi 2018 gilvvohallamii. Dát lea stuorámus mearri oasseválddiin ja seamma mearri go lei jagiin 2008 ja 2011 gilvvohallamiin.	Forty-three countries participate in the contest, equalling the record of the 2008 and 2011 editions.	Forty-three countries participate in the contest, equalling the record of the 2008 and 2011 editions.	Both

Figure 6.6: Examples in which the sentence representations the language model generates are influenced by numbers.

Index	Language	Source text	True match	Predicted match	Correct setup
1	Icelandic	Hvernig komst hugmyndin um ódauðlega sál inn í hina " kristnu " kenningu?	How , in fact , did the immortal soul idea get into “ Christian ” teaching ?	What do you believe about man’s supposed immortality ?	high-dttr
2	Icelandic	Við erum hér að fást við hugarfóstur dauðans, ekki dauðann sjálfan.	We are dealing here with the fantasy of death , not with death itself .	The Bible translator William Tyndale (c . 1492 - 1536) wrote in the foreword to his translation : “ In putting departed souls in heaven , hell , or purgatory you destroy the arguments wherewith Christ and Paul prove the resurrection . . . *or:* His death would atone for man’s sins.	Neither
3	Meadow Mari	Мутланымашым тӱҥалше каласымашым налме	Message received begins conversation	Calm *or:* Co_nversations:	Neither

Figure 6.7: Examples in which the sentence representations the language model generates are influenced by matching phrases.

sensitivity of the model representations to this clue increases, however, when the Arabic language adapter is stacked. The candidate translation becomes *Happy new year 2015 – What’s in store?*, matching the source sentence closer both in length and shared content. However, this does not guarantee success.

Finally, it appears that the presence of numbers does not always influence model representations in a meaningful way. In example 3, neither top (evaluation based on $k = 1$) candidate translations match the source sentence, despite the presence of the *25 percent* and the year *1983*. It seems that here the presence of the additional language adapter makes a difference, possibly by making the sentence representations more sensitive to numbers. When the Vietnamese language adapter is stacked on top of the Icelandic one, the correct translation is returned at least within the top three candidates.

Overlapping meaning

Figure 6.7 shows examples of overlapping semantic content between the source sentence and candidate translations. In some cases, this in itself does not guarantee an accurate

match though. These are the types of errors that are more common with Icelandic than Amharic, Meadow Mari, and Northern Sami. This reflects the difference in model knowledge between a low resource language, Icelandic, that the multilingual BERT model was exposed to during its pretraining, versus zero-shot languages that the model has much less knowledge on.

Even if the overall scores do not improve, it does appear that stacked language adapters can contribute to the model knowledge on the semantics of specific languages using cross-lingual transfer.

In example 1, for instance, only using the Icelandic language adapter results in retrieving the candidate translation with the wrong meaning. However, both sentences share the concept of *immortality*. Adding the Vietnamese language adapter, while overall hurting the performance on the Icelandic test set, improves model knowledge on this concept, returning the correct translation. In example 2, the shared concept between the source sentence and the candidate sentences is *death*. However, when using only the target language adapter, the connection between the sentences is more abstract. The link becomes more specific, though not necessarily more accurate, when the Vietnamese language adapter is stacked on top of the Icelandic one.

Finally, in example 3, it appears that the multilingual BERT model does not have adequate knowledge on Meadow Mari to create well-aligned representations with English, even when the Meadow Mari language adapter is applied. However, adding the English language adapter on top of the Meadow Mari one results in a sentence representation that matches English *conversations*, even if it does not match the true translation.

Examples like these are not too common for Meadow Mari, but at least exist. It is likely that the comparatively larger training dataset for Meadow Mari – 2.6 million words as opposed to 1.7 million words for Amharic and 690,000 for Northern Sami – results in more robust representations that allow the language model to grasp a deeper, more semantic content in Meadow Mari sentences (see Table 6.2).

6.2.2 Analysing factors

After a qualitative analysis on the level of individual examples in the previous section, in this section I return to a more quantitative analysis, concentrating on the zero-shot languages in my sample, Amharic, Meadow Mari, and Northern Sami. I investigate the contribution of stacked language adapters based on a number of factors related to characteristics of the source sentences of the cross-lingual sentence retrieval task. In this, I follow principles of the analysis proposed in the XTREME-R benchmark (Ruder et al., 2021), where they use ExplainaBoard (Liu et al., 2021a) to augment analysis of the Tatoeba (cross-lingual sentence retrieval) task.

The main principle of ExplainaBoard is to analyse the characteristics of the examples in the test dataset by selecting a factor, assigning (*bucketing*) the data into distinct categories based on this factor, and observing how the performance might be affected on these categories. In my own investigation, I select novel factors not covered by ExplainaBoard that I believe might be relevant for the cross-lingual sentence retrieval task and the combination of language adapters.

I focus on two factors that I believe might affect on how language adapter stacks impact individual examples: the coverage of the subwords in the original model vocabulary on the source sentence, and the amount of UNK (*unknown*) tokens that appear in the source sentence.

I approximate the coverage of the subwords in the model vocabulary in terms of *fertility*. I also define fertility in Section 2.3.1 following Ács (2019): it stands for the average number of subword tokens generated from a single 'real' token. For individual sentences, I calculate fertility by dividing the number of subword tokens in a source sentence by the number of real tokens. If this number is high, that might reflect that the subwords in the original model vocabulary does not cover the source sentence well.

UNK tokens also measure the coverage of the model vocabulary. When the multilingual BERT model encounters with characters in a token that were missing in its pretraining data, it replaces the token with an UNK token. This number is especially high for Amharic, as its Ge'ez script was not seen by the model during its pretraining. I measure the amount of UNK tokens by using the ratio of UNK tokens versus total count of subword tokens in a sentence. Since Latin characters are generally well-covered in the multilingual BERT training data, Northern Sami, that is written with the Latin alphabet does not feature UNK tokens. Thus this analysis can only be carried out for Amharic and Meadow Mari, the two zero-shot languages that are not written with the Latin script.

Source text fertility

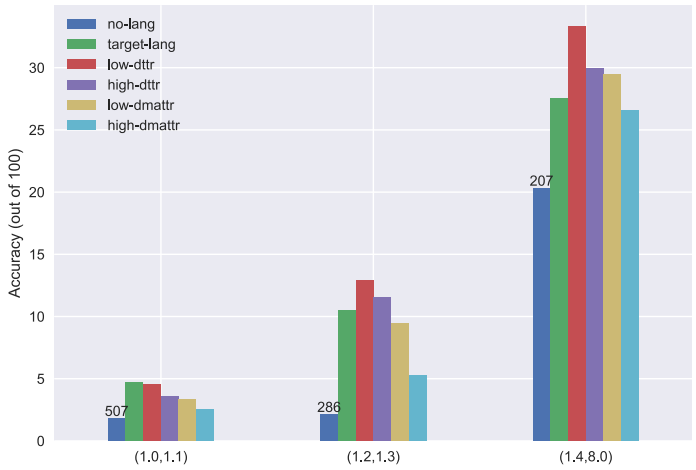
Figure 6.8 shows the accuracy scores per source text fertility for Amharic (graph in 6.8a), Meadow Mari (graph in 6.8b), and Northern Sami (graph in 6.8c). The sentences in the test set are arranged into buckets depending on their fertility values, and performance is measured for the individual buckets. In the graphs, the x axis shows the buckets in terms of inclusive intervals of the fertility values.

As mentioned above, fertility can be interpreted as the coverage of the tokenizer of the multilingual BERT model in a given language. The higher this fertility value is, the more subword tokens the input is broken down into. Breaking an input word into multiple subword tokens might be appropriate when the input word has a complex internal structure and the subword tokens at least broadly correspond to meaningful linguistic subdivisons.

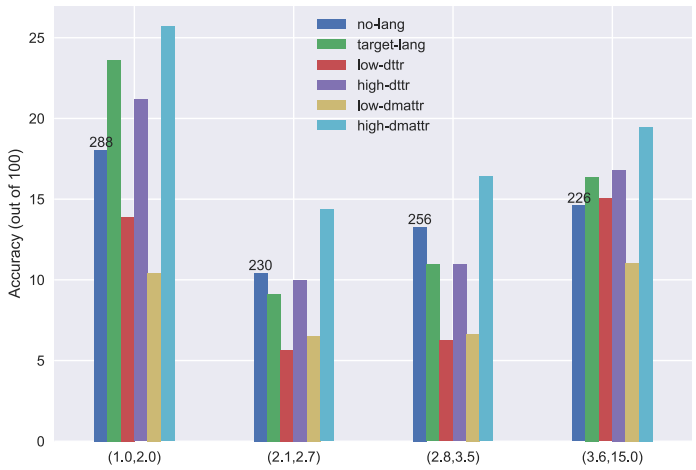
In practice, however, breaking down an input word into numerous subword tokens often rather indicates that the multilingual BERT model cannot appropriately represent this particular input word with items in its vocabulary, and has to roll back to shorter, or in fact character-level, representations. This could hinder the model ability to appropriately represent meaning in its subword elements, thus making it more difficult for it to create meaningful sentence-level representations. This is why we might expect performance to be lower in general on sentences with higher fertility values.

This expectation is supported by the results on Meadow Mari and Northern Sami. For these languages, performance generally decreases as the fertility increases, but the relationship is not linear. The highest scores are achieved all across the board by using English as a stacked adapter in the **high-dmattr** setting. Top performance for most experiments is reached when the fertility of the sentences is generally low, between 1.0 and 2.0. However, for Meadow Mari, the lowest performance is reached when the fertility is medium low, between 2.1 and 2.7, while for Northern Sami, the performance on the sentences with higher fertility is relatively even depending on the experiment.

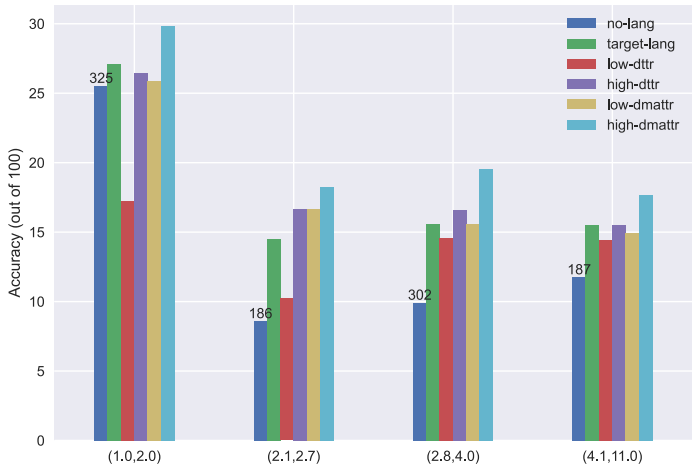
For Meadow Mari, the highest contribution of the target language adapter is on sentences with relatively low fertility values. In the other buckets, it might in fact hurt performance or barely contribute. All experimental setups show similar trends otherwise, except that for experiments involving a stacked language adapter with a



(a) Accuracy scores per source text fertility per experiment (Amharic)



(b) Accuracy scores per source text fertility per experiment (Meadow Mari)



(c) Accuracy scores per source text fertility per experiment (Northern Sami)

Figure 6.8: Performance on source text fertility buckets for different experiments. Numbers in the graph show the count of sentences that belong to a specific bucket.

low distance in terms of type-token ratio (**low-dttr** and **low-dmattr**), performance is higher when the fertility scores are between 3.6 and 15.0 then when they are closer to 1.0 and 2.0. For Northern Sami, the trends are similar as for Meadow Mari, but perhaps with less extreme differences.

The graph in 6.8a shows that the trend is different for Amharic, where the higher fertility scores correspond to higher performance, especially when language adapters are used. The most likely explanation for this is that since the Ge'ez script Amharic uses was not seen by the multilingual BERT model during training, a large proportion of Amharic texts is dominated by UNK tokens. This also comes across in the graph representing the ratio of UNK tokens for Amharic in 6.9a. It appears that the contribution of the use of language adapters and their combinations is most significant on the middle bucket, when the ratio between subword tokens and real word tokens is between 1.2 and 1.3. The contribution of the Arabic language adapter in the **low-dttr** setting is the largest compared to the **target-lang** setup is on the top bucket (fertility of 1.4-8.0), while it hurts performance on the lowest bucket (fertility of 1.0-1.1). Otherwise, trends between the different experiments stay constant.

Ratio of UNK tokens in the source text

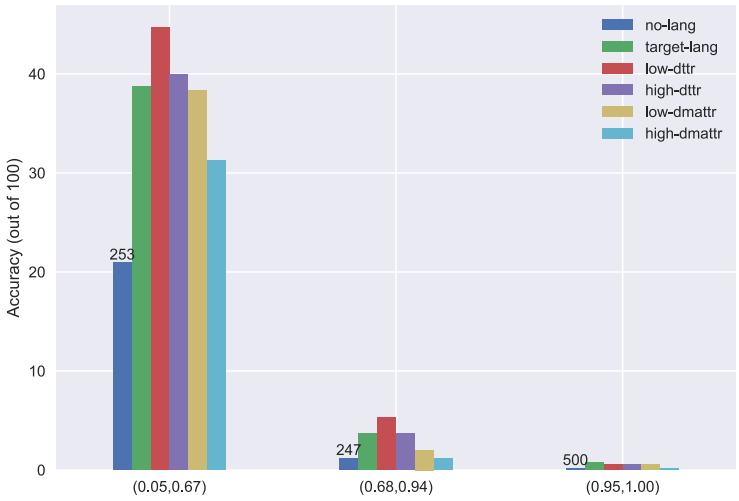
Figure 6.9 shows the accuracy scores per the ratio of UNK tokens for Amharic (6.9a) and Meadow Mari (6.9b). In both languages there are characters that the multilingual BERT model has not encountered during its pretraining. When a word has characters that the model is not familiar with, it replaces them with an UNK (*unknown*) token. This means that the model cannot initialise the contextual embedding of such token from a pretrained subword representation stored in its embedding matrix of the model, likely making it more difficult to create meaningful embeddings for it. This makes it reasonable to assume that the higher the ratio of UNK tokens is in a sentence, the less likely a good sentence representation can be generated for it, thus the worse the accuracy is going to be on the cross-lingual sentence retrieval task.

This assumption is supported by both Figure 6.9a and Figure 6.9b. Performance on sentences with a lower ratio of UNK tokens is consistently higher for both Amharic and Meadow Mari.

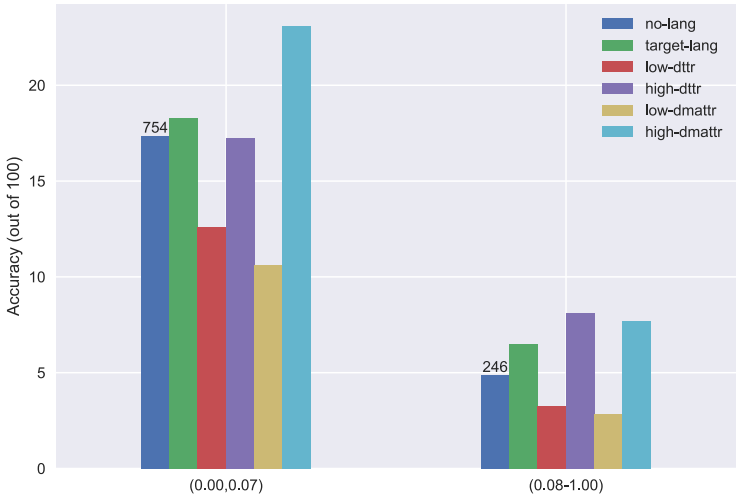
This is especially clear with Amharic which is written with a script multilingual BERT has not seen during its training. On examples where the ratio of UNK tokens is relatively low, between 0.05 and 0.67, performance on cross-lingual sentence retrieval reaches close to 40% accuracy with the **target-lang** baseline and the **high-ttr** adapter combination (with Vietnamese), and it far surpasses 40% when stacking the Arabic language adapter on top of the Amharic one (**low-dttr** setup). All these setups far surpass the performance achieved with using no language adapters (**no-lang**). On a similar-size group of sentences with UNK token ratios of 0.68-0.94, the relative performance of the different experiments is the same as in the previous group, the most marked difference being that the accuracy scores are much lower, barely even reaching 5%.

Half the test set, 500 sentences, belong to the group with the highest UNK token ratios between 0.95 and 1.00. In the case of sentences with a ratio of 1.00, that means that every token in the input sentence is replaced with an UNK token. It is not surprising that in these cases, performance barely goes above 0, accounting for the low performance on Amharic.

In the case of Meadow Mari, test sentences belong to only two buckets with respect to UNK token ratio. Three-quarters of the sentences belongs to the bucket with UNK



(a) Accuracy scores per the ratio of UNK tokens in source text per experiment (Amharic)



(b) Accuracy scores per the ratio of UNK tokens in source text per experiment (Meadow Mari)

Figure 6.9: Performance on the ratio of UNK tokens in the source text in buckets for different experiments. Numbers in the graph show the count of sentences that belong to a specific bucket.

token ratios between 0.00 and 0.07. Performance is higher on these sentences than on the remaining quarter with UNK token ratios between 0.08 and 1.00. On sentences with low UNK token ratios the highest performance is reached by stacking the English language adapter in the **high-dmattr** setup. This might be due to the contribution of English in achieving better alignment on cases where there is Latin characters mixed in with the Meadow Mari text.

However, on sentences with higher UNK token ratios, the performance is slightly higher when the Vietnamese language adapter is stacked on top of the Meadow Mari adapter in the **high-dttr** setup. This further illustrates the perplexingly positive contribution of Vietnamese.

6.3 Future work

The size and architecture of large pretrained language models such as multilingual BERT makes it difficult to interpret what they learn during pretraining, and how they apply this knowledge during their use. Investigating how such language models transfer information across languages is similarly a challenging task. I believe that in my thesis, I made some interesting discoveries in how stacked language adapters might facilitate cross-lingual transfer across languages. This is a novel research direction and I believe I have shown that there are numerous insights that can be drawn from such approach.

There are a number of unexpected findings that I could not satisfactorily answer. In future work, I would focus on investigating why language adapters of certain languages, such as Vietnamese, seem to induce cross-lingual transfer across languages. There are also other languages that seem to harm multilingual BERT knowledge on even typologically similar languages, for instance Finnish and Estonian. It is also worth probing why this is the case. Finally, I would like to carry out more work to test methodically how appropriate it is to reduce languages to a single typological feature which appears to be widespread practice in related work.

In the following, I provide additional suggestions on continuing my research direction, also addressing certain aspects of this thesis that can be improved.

6.3.1 Alternative evaluation methods

In my work, I evaluate the success of cross-lingual transfer between language adapters using a cross-lingual sentence retrieval task. Besides its relevance for machine translation and for TAUS, the company I carry out my thesis with, the other advantage of the task is that it has ample datasets available that allow evaluation on even very low resource languages. However, in future work, it might be interesting to evaluate the success of cross-lingual transfer using a series of tasks in conjunction with cross-lingual sentence retrieval. Tasks are especially valuable if the test sets are parallel, so unlike in my own work, scores can directly be compared across languages.

The XTREME (Hu et al., 2020) and XTREME-R (Ruder et al., 2021) benchmarks from where the cross-lingual sentence retrieval task is adopted contain a series of tasks that allow cross-lingual comparison (see Section 3.3). The disadvantage is that it is difficult to achieve good coverage with them on low resource languages, the ones that cross-lingual transfer using stacked language adapters might actually benefit (see Section 6.1). There are other datasets that cover low resource languages, for instance MasakhaNER for a set of 10 African languages for the task of named entity recognition

(Adelani et al., 2021). The drawback is that to cover these languages, we would need to train new language adapters.

Alternatively, cross-lingual transfer can also be measured on a specific language modelling objective. As mentioned in Section 2.1.1, language models learn to assign probabilities to input sequences of language. If the language model is capable of assigning high probability to sentences of a language, that indicates it has no difficulty modelling that language, i.e., it has linguistic knowledge of it. When parallel corpora are used to measure the language modelling difficulty of different languages, we can compare the model knowledge on these languages (see Section 2.3.4 for a discussion on studies that use this method). If a particular language adapter combination increases the probability the language model assigns to a text in a particular language, we can deduct that the language model knowledge increased on that language.

One dataset to use to compare the language modelling difficulty of different languages is the Parallel Bible Corpus (Mayer and Cysouw, 2014) covering 800 languages. In this thesis, I use it to measure morphological complexity. While other massively multilingual parallel corpora are not easy to come by, there are resources such as MulTED, an aligned and part-of-speech tagged parallel corpus covering 101 languages, sourced from subtitles of TED talks (Zeroual and Lakhouaja, 2022).

Another way of measuring how language model knowledge changes due to cross-lingual transfer on different languages is to directly measure the isomorphism of the embedding spaces within the shared embedding space of a multilingual language model. Jones et al. (2021) investigates this isomorphism with three different measures on a Bible corpus (Christodouloupoulos and Steedman, 2015). By investigating how the inter-lingual isomorphism changes with language adapter combinations, we can gain insight into how language model knowledge is affected through different combinations.

6.3.2 Alternate adapter combinations

In my thesis, I investigate how cross-lingual transfer between stacked language adapters is affected by the similarity in terms of a single typological feature, morphological complexity as quantified by type-token ratio.

There are other possibilities. Other typological features can be selected from resources such as WALS (Dryer and Haspelmath, 2013), and can be used as a basis on which languages can be combined (see Section 2.3.2). These can include syntactic features, such as word order, or morphological features, for example the exponence of tense-aspect-mood inflection, or the locus of marking in possessive noun phrases (Park et al., 2021).

However, selecting one, or only a few, WALS features follows a naive assumption that languages can be reduced to a single typological feature where in fact it appears that parameter sharing across typological similarity is a more complex phenomenon. Additionally, lower resource languages might have missing values for WALS features, making coverage difficult for these.

Another option is to use typological language vectors, such as lang2vec, that represent individual languages across dimensions of syntax, phonology, and phonetic inventory (Littell et al., 2017). These vectors are likely to provide a more accurate representation of the typological features of languages, but they might have missing values for lower resource languages. Additionally, due to the large number of features lang2vec vectors represent, isolating individual factors behind cross-lingual transfer might be challenging.

Another factor that might be changed in future work is how adapters are combined. In this research, I limit myself to stacking language adapters on top of each other, not using any mechanisms to specifically induce cross-lingual transfer. This approach has the advantage of simplicity, however, previous work describes alternative methods of combining knowledge inside adapters. AdapterFusion, for instance, allows explicit knowledge composition between adapters, preventing interference between the information stored in various adapters (Pfeiffer et al., 2021a). This might preclude performance to be harmed by the second language adapter at all.

It is also possible that different language model adaptation methods could complement each other. In future experiments, it might be interesting to combine language adapters with other approaches, such as retraining the embedding layer of the language model on target language data while freezing the other model layers (Pfeiffer et al., 2021b). In this case, adaptation to the target language would take place on a lower level in the model, while further model layers would allow cross-lingual transfer by the language adapter of the second language.

6.3.3 Training new adapters

Besides the pretraining data size of language adapters, there are other factors at play in how cross-lingual transfer is affected by stacked language adapters that seems to play a larger role. However, it is still worth investigating the interplay of the pretraining data sizes used to train a language adapter and its contribution to the performance when stacked on top of the target language adapter. Creating new language adapters while controlling for the size of the pretraining data could help in isolating factors stemming from how well-trained a particular adapter is.

It is also possible that the cross-lingual transfer capabilities of language adapters could be positively affected by using pretraining data from multiple languages to create so-called *bilingual* language adapters. Bapna and Firat (2019) create such bilingual language adapters for language pairs in the context of neural machine translation. Their approach enhances performance on machine translation, which means they contribute to making representations of the languages they combine more aligned. Suppose language a is the low-resource or zero-shot target language, and language b is a language that the multilingual language model has been exposed to in its pretraining. It is possible that a bilingual adapter for the two languages could make the model representations for language a more aligned with those of language b. This might in turn help the language model in leveraging its knowledge on language b to process input from language a.

Chapter 7

Conclusion

In this thesis, I investigated how cross-lingual transfer can be induced within pretrained massively multilingual language models using stacked language adapters with the goal to improve model knowledge on low resource languages. I carried out 80 experiments for a diverse set of 8 languages, assessing the finding in the literature that cross-lingual transfer is most significant between languages that are similar to each other in terms of typological features. To establish this similarity, I used corpus-driven measures of morphological complexity estimated from a parallel Bible corpus. I compared accuracy scores on a cross-lingual sentence retrieval (or translation pair detection) task between two baseline experiments, as well as different adapter stacks matching languages with both low and high distance scores in terms of morphological complexity.

I have shown that certain language adapter stacks can improve performance on the evaluation task for low resource languages, indicating an increase in model knowledge and successful cross-lingual transfer. I have shown that this is partially because the stacked language adapter enhances the sensitivity of the language model to certain cues in the input that it seems to pay attention to, including the presence and quality of special characters and numbers.

I have also demonstrated that similarity between languages measured in terms overall type-token ratio and moving average type-token ratio – the corpus-driven morphological complexity measures I used in this thesis – does not seem to correlate with whether a certain adapter stack will increase performance on a language. Either these measures are not appropriate for establishing typological similarity between languages, or there are other factors that may facilitate or hinder cross-lingual transfer between stacked language adapters. Further investigation is needed to isolate what linguistic and extra-linguistic factors might contribute to positive transfer. In this work, I discussed factors such as the pretraining data size of different languages, lexical overlap and shared script, but none of them seem to correlate with cross-lingual transfer. It is even possible that extant practice in similar research in reducing languages to a single feature, or a small selection of features, might be too simplifying an assumption when comparing languages.

My main contribution is showing that cross-lingual transfer can be investigated using stacked language adapters. This is a flexible and straightforward approach that does not require additional training, making it possible to incorporate a large number of linguistic and extra-linguistic factors across a wide range of languages. While there are open questions, my thesis presented a promising methodology that can be easily followed up in future work.

Appendix A

Full results

Language	no-lang	target-lang	low-dttr	high-dttr	low-dmattr	high-dmattr
Amharic	5.7	11.1	12.9	11.3	10.5	8.3
Meadow Mari	14.3	15.4	10.3	15.0	8.7	19.3
French	60.3	53.0	25.7	48.6	22.8	12.3
Hungarian	32.0	44.8	13.8	36.4	19.7	37.3
Indonesian	48.3	57.1	22.6	51.9	32.7	22.0
Icelandic	20.2	38.6	29.8	40.2	19.8	16.7
Northern Sami	15.1	19.1	14.6	19.6	19.0	22.3
Vietnamese	50.3	62.2	54.7	51.8	33.6	26.3

Table A.1: Accuracy scores on the cross-lingual sentence retrieval task per experiment for each language in the sample.

Language	low-dttr	high-dttr	low-dmattr	high-dmattr
Amharic	+16.22%	+1.80%	−5.41%	−25.23%
Meadow Mari	−33.12%	−2.60%	−43.51%	+25.32%
French	−51.51%	−8.30%	−56.98%	−76.79%
Hungarian	−69.20%	−18.75%	−56.03%	−16.74%
Indonesian	−60.42%	−9.11%	−42.73%	−61.47%
Icelandic	−22.80%	+4.15%	−48.70%	−56.74%
Northern Sami	−23.56%	+2.62%	−0.52%	+16.75%
Vietnamese	−12.06%	−16.72%	−45.98%	−57.72%

Table A.2: Change in accuracy in percentages compared to **target-lang** baseline for each adapter stack.

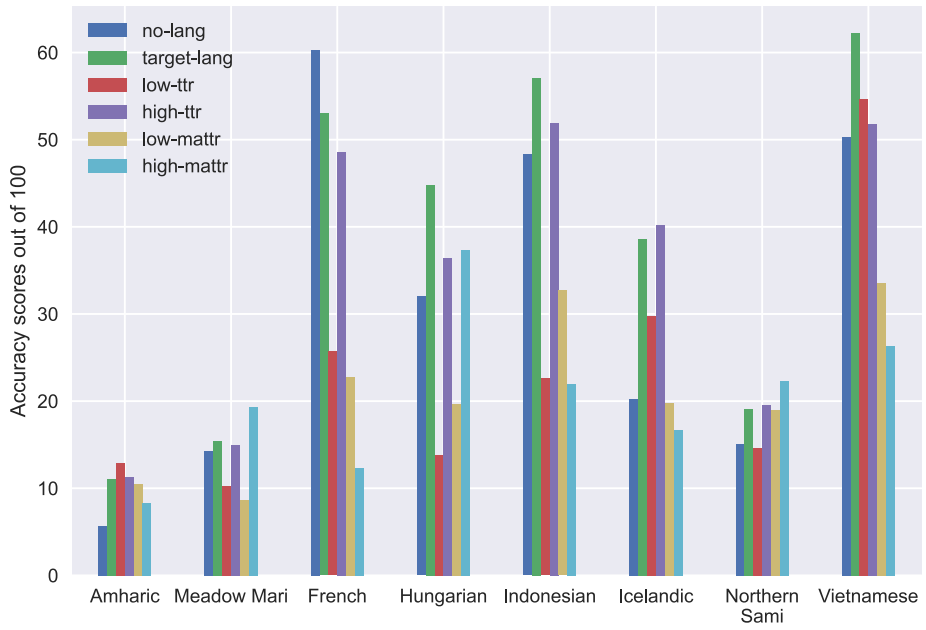


Figure A.1: Accuracy scores on the cross-lingual sentence retrieval task per experiment for each language in the sample.

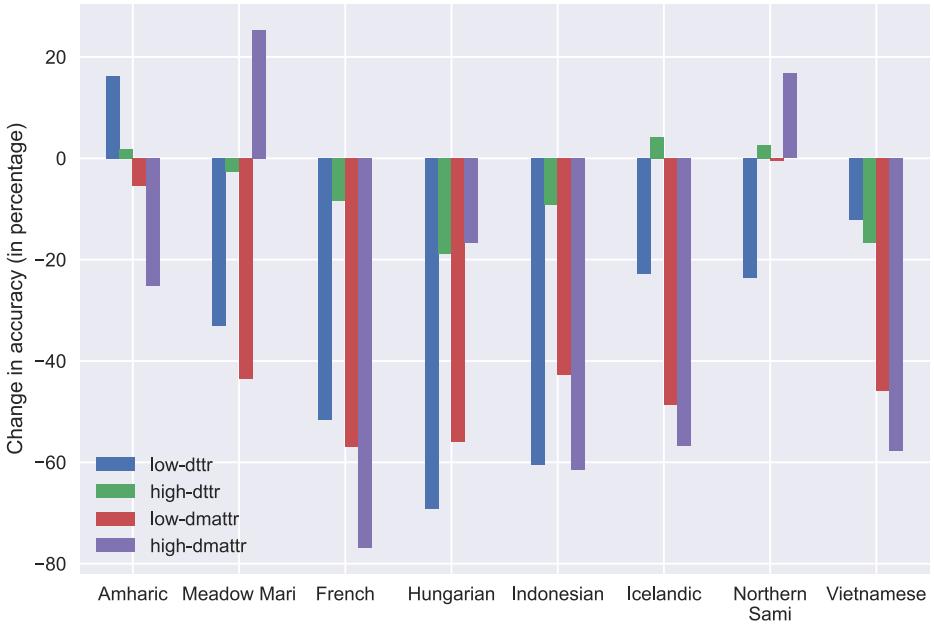


Figure A.2: Change in accuracy in percentages compared to **target-lang** baseline for each adapter stack.

Appendix B

Additional material

Adapter order	low-dttr	high-dttr	low-dmattr	high-dmattr
Amharic adapter first	**12.9**	11.3	10.5	8.3
Amharic adapter second	**11.9**	11.6	11.5	8.6
French adapter first	25.7	**48.6**	22.8	12.3
French adapter second	27.8	**44.5**	21.7	16.2
Meadow Mari adapter first	10.3	**15.0**	8.7	19.3
Meadow Mari adapter second	7.1	**13.1**	8.2	18.1
Hungarian adapter first	13.8	36.4	19.7	**37.3**
Hungarian adapter second	14.1	33.7	16.9	**34.8**
Indonesian adapter first	22.6	**51.9**	32.7	22.0
Indonesian adapter second	18.9	**49.0**	29.7	15.2
Icelandic adapter first	29.8	**40.2**	19.8	16.7
Icelandic adapter second	27.8	**38.1**	16.3	15.6
Northern Sami adapter first	14.6	19.6	19.0	**22.3**
Northern Sami adapter second	14.2	18.5	18.4	**20.2**
Vietnamese adapter first	**54.7**	51.8	33.6	26.3
Vietnamese adapter second	49.3	**49.5**	28.3	19.7

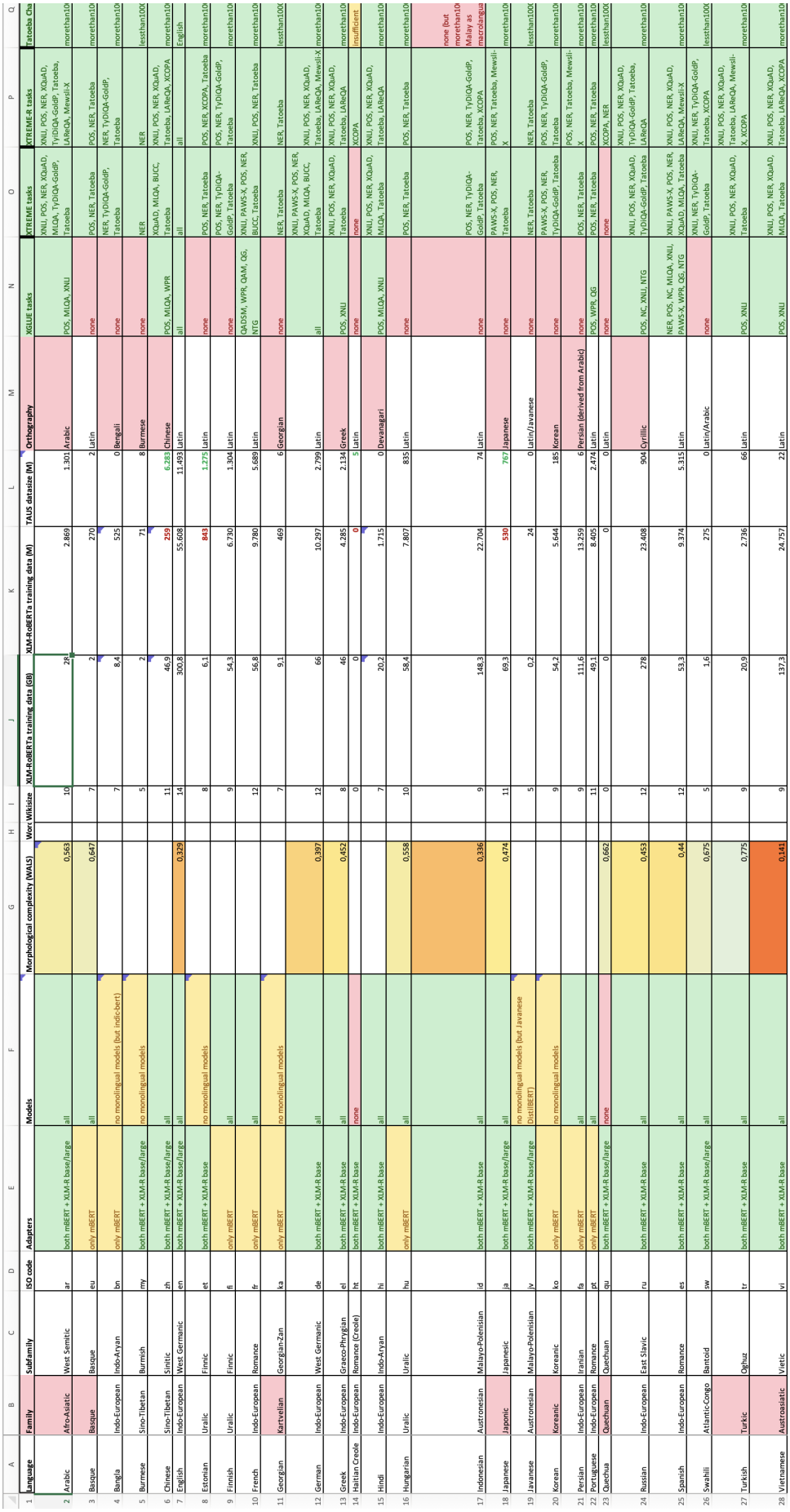
Table B.1: Difference in accuracy scores between adapter stacks where the target language adapter is the first or second on in the stack. The highest scores in each category are highlighted with bold font, while the lowest scores are underlined. It can be seen that while the scores fluctuate, the trends are not significantly changed by the order of the language adapters.

Language name	TTR	MATTR	Line count	No. of Bibles	TTR Variance	MATTR Variance
Amharic	0.212	0.628	7656	1		
Arabic	0.188	0.650	7959	1		
Armenian	0.114	0.543	7958	1		
Basque	0.124	0.627	7853	2	9.00E-05	2.00E-05
Bangla	0.050	0.519	7872	5	5.00E-05	5.00E-05
Burmese	0.458	0.644	7766	2	0.04078	0.01168
Buryat	0.131	0.668	7827	1		
Czech	0.121	0.602	7953	6	0.00012	0.00051
English	0.034	0.410	7941	27	6.00E-05	0.00036
Erzya	0.117	0.608	7955	1		
Estonian	0.107	0.537	7855	1		
Finnish	0.136	0.596	7957	3	0.00035	0.00184
French	0.055	0.453	8172	16	5.00E-05	0.00031
Georgian	0.153	0.573	4894	1		
German	0.061	0.484	8310	22	4.00E-05	0.00053
Greek	0.108	0.510	7942	3	0.00049	0.00094
Haitian Creole	0.014	0.328	7953	2	0	0
Hungarian	0.144	0.567	7930	5	0.00017	0.00031
Icelandic	0.084	0.507	7860	1		
Ilocano	0.087	0.424	7938	1		
Indonesian	0.042	0.445	7936	5	1.00E-05	0.00017
Javanese	0.066	0.467	7959	2	0	0
Korean	0.180	0.696	7850	4	0.00028	0.00059
Latin	0.119	0.581	12509	2	0	4.00E-05
Latvian	0.114	0.552	7951	2	2.00E-05	0
Meadow Mari	0.104	0.606	7946	1		
Maori	0.014	0.297	7957	1		
Northern Sami	0.097	0.540	7951	1		
Persian	0.067	0.481	7900	3	0.00016	9.00E-05
Portuguese	0.064	0.479	7918	6	0.00016	0.00015
Russian	0.123	0.580	7906	6	0.0001	4.00E-05
Spanish	0.061	0.458	7754	6	0.0001	0.00013
Swahili	0.120	0.55	7740	3	4.00E-05	2.00E-05
Tamil	0.221	0.700	7874	1		
Turkish	0.169	0.662	7442	3	0.0001	5.80E-04
Vietnamese	0.017	0.422	7903	6	1.00E-05	0.00016
Wolof	0.037	0.461	7959	1		

Table B.2: Full list of overall type-token ratio (TTR) and moving average type-token ratio (MATTR) scores derived from the Parallel Bible Corpus. The exact count of Bible versions is listed for each language in the **No. of Bibles** column. If there were more than one Bible versions available for a language, the **TTR, MATTR** and **Line count** columns are averages of all available Bible versions, and I provide the variances between the scores as well. Variance between different Bible versions in general is very low, showing that the content is parallel enough across versions.

Language	**low-dttr** language	**low-dttr** distance	**high-dttr** language	**high-dttr** distance	**low-dmattr** language	**low-dmattr** distance	**high-dmattr** language	**high-dmattr** distance
Amharic	Arabic	0.0163	Vietnamese	131.57439	Burmese	0.00062	English	0.28271
Arabic	Korean	0.00198	Vietnamese	101.17993	Burmese	9e-05	English	0.34265
Armenian	Erzya	0.00066	Vietnamese	32.55709	Northern Sami	3e-05	English	0.10523
Basque	Russian	7e-05	Vietnamese	39.61592	Burmese	0.0007	English	0.28012
Bengali	French	0.00826	Vietnamese	3.76817	Greek	0.00031	English	0.07068
Burmese	Amharic	1.34648	Vietnamese	672.94464	Arabic	9e-05	English	0.32573
Buryat	Finnish	0.00135	Vietnamese	44.96886	Turkish	8e-05	English	0.39598
Czech	Swahili	7e-05	Vietnamese	37.42561	Meadow Mari	4e-05	English	0.2193
English	Wolof	0.00657	Vietnamese	1.0	Vietnamese	0.00081	Korean	0.16885
Erzya	Latin	0.00028	Vietnamese	34.60208	Meadow Mari	1e-05	English	0.23322
Estonian	Greek	9e-05	Vietnamese	28.02768	Northern Sami	3e-05	English	0.09595
Finnish	Buryat	0.00146	Vietnamese	49.0	Czech	0.0001	English	0.20581
French	German	0.00967	Vietnamese	4.99654	Spanish	0.00012	Korean	0.1219
German	Portuguese	0.0022	Vietnamese	6.69896	Persian	4e-05	Korean	0.09278
Greek	Estonian	9e-05	Vietnamese	28.65398	Icelandic	4e-05	Korean	0.07142
Hungarian	Finnish	0.00346	Vietnamese	55.80969	Russian	0.0005	English	0.14663
Icelandic	Ilocano	0.00119	Vietnamese	15.53287	Greek	3e-05	Korean	0.07374
Ilocano	Icelandic	0.00128	Vietnamese	16.95502	Vietnamese	2e-05	Korean	0.15273
Indonesian	Wolof	0.01826	Vietnamese	2.16263	French	0.00031	Korean	0.13006
Javanese	Persian	0.00022	Vietnamese	8.30796	Wolof	0.00017	Korean	0.10826
Korean	Arabic	0.00181	Vietnamese	91.93426	Buryat	0.00176	English	0.48659
Latin	Swahili	7e-05	Vietnamese	36.0	Hungarian	0.00061	English	0.17395
Latvian	Erzya	0.00066	Vietnamese	32.55709	Swahili	1e-05	English	0.11995
Meadow Mari	Estonian	0.00079	Vietnamese	26.19031	Erzya	1e-05	English	0.22853
Northern Sami	Meadow Mari	0.00453	Vietnamese	22.14533	Armenian	3e-05	English	0.10054
Persian	Javanese	0.00023	Vietnamese	8.65052	Portuguese	2e-05	Korean	0.09542
Portuguese	Javanese	0.00092	Vietnamese	7.6436	Persian	2e-05	Korean	0.09721
Russian	Basque	7e-05	Vietnamese	38.87889	Hungarian	0.00053	English	0.17192
Spanish	Portuguese	0.0022	Vietnamese	6.69896	Wolof	4e-05	Korean	0.11693
Swahili	Czech	7e-05	Vietnamese	36.70934	Latvian	1e-05	English	0.1166
Turkish	Quechua	3e-05	Vietnamese	79.94464	Buryat	8e-05	English	0.37778
Vietnamese	English	0.25	Burmese	0.92714	Ilocano	2e-05	Korean	0.15498
Wolof	English	0.00779	Vietnamese	1.38408	Spanish	4e-05	Korean	0.114

Table B.3: Languages with the lowest and highest distance scores in terms of overall and moving average type-token ratios.



Language	Family	Subfamily	ISO code	Adapters	Models	Morphological complexity (WALS)	Wori Wikisize	XLM-RoBERTa training data (GB)	XLM-RoBERTa training data (M)	TAUS datasize (M)	Orthography	XGLUE tasks	XTREME tasks	XTREME-R tasks	Tatoeba Che
Arabic	Afro-Asiatic	West Semitic	ar	both mBERT + XLM-R base/large	all	0,563	10	28	2.869	1.301	Arabic	POS, MLQA, XNLI	XNLI, POS, NER, XQuAD, MLQA, TyDiQA-GoldP, Tatoeba	XNLI, POS, NER, XQuAD, TyDiQA-GoldP, Tatoeba, LAReQA, Mewsli-X	morethan10
Basque	Basque	Basque	eu	only mBERT	all	0,647	7	2	270	2	Latin	none	POS, NER, Tatoeba	POS, NER, Tatoeba	morethan10
Bangla	Indo-European	Indo-Aryan	bn	only mBERT	no monolingual models (but indic-bert)		7	8,4	525	0	Bengali	none	NER, TyDiQA-GoldP, Tatoeba	NER, TyDiQA-GoldP, Tatoeba	morethan10
Burmese	Sino-Tibetan	Burmish	my	both mBERT + XLM-R base/large	no monolingual models		5	2	71	8	Burmese	none	NER	NER	lessthan100
Chinese	Sino-Tibetan	Sinitic	zh	both mBERT + XLM-R base/large	all		11	46,9	259	6.283	Chinese	POS, MLQA, WPR	XQuAD, MLQA, BUCC, Tatoeba	XNLI, POS, NER, XQuAD, Tatoeba, LAReQA, XCOPA	morethan10
English	Indo-European	West Germanic	en	both mBERT + XLM-R base/large	all	0,329	14	300,8	55.608	11.493	Latin	all	all	all	English
Estonian	Uralic	Finnic	et	both mBERT + XLM-R base	no monolingual models		8	6,1	843	1.275	Latin	none	POS, NER, Tatoeba	POS, NER, XCOPA, Tatoeba	morethan10
Finnish	Uralic	Finnic	fi	only mBERT	all		9	54,3	6.730	1.304	Latin	none	POS, NER, TyDiQA-GoldP, Tatoeba	POS, NER, TyDiQA-GoldP, Tatoeba	morethan10
French	Indo-European	Romance	fr	only mBERT	all		12	56,8	9.780	5.689	Latin	QA(D)SM, WPR, QAM, QG, NTG	XNLI, PAWS-X, POS, NER, BUCC, Tatoeba	XNLI, POS, NER, Tatoeba	morethan10
Georgian	Kartvelian	Georgian-Zan	ka	only mBERT	no monolingual models		7	9,1	469	6	Georgian	none	NER, Tatoeba	NER, Tatoeba	lessthan100
German	Indo-European	West Germanic	de	both mBERT + XLM-R base	all	0,397	12	66	10.297	2.799	Latin	all	XNLI, PAWS-X, POS, NER, XQuAD, MLQA, BUCC, Tatoeba	XNLI, POS, NER, XQuAD, Tatoeba, LAReQA, Mewsli-X	morethan10
Greek	Indo-European	Graeco-Phrygian	el	both mBERT + XLM-R base	all	0,452	8	46	4.285	2.134	Greek	POS, XNLI	XNLI, POS, NER, XQuAD, Tatoeba	XNLI, POS, NER, XQuAD, Tatoeba, LAReQA	morethan10
Haitian Creole	Indo-European	Romance (Creole)	ht	both mBERT + XLM-R base	none		0	0	0	5	latin	none	none	XCOPA	insufficient
Hindi	Indo-European	Indo-Aryan	hi	both mBERT + XLM-R base	all		7	20,2	1.715	0	Devanagari	POS, MLQA, XNLI	XNLI, POS, NER, XQuAD, MLQA, Tatoeba	XNLI, POS, NER, XQuAD, Tatoeba, LAReQA	morethan10
Hungarian	Uralic	Uralic	hu	only mBERT	all	0,558	10	58,4	7.807	835	Latin	none	POS, NER, Tatoeba	POS, NER, Tatoeba	morethan10
Indonesian	Austronesian	Malayo-Polenisian	id	both mBERT + XLM-R base	all	0,336	9	148,3	22.704	74	Latin	none	POS, NER, TyDiQA-GoldP, Tatoeba	POS, NER, TyDiQA-GoldP, Tatoeba, XCOPA	none (but morethan10 Malay as macrolangua
Japanese	Japonic	Japanesic	ja	both mBERT + XLM-R base/large	all	0,474	11	69,3	530	767	Japanese	none	PAWS-X, POS, NER, Tatoeba	POS, NER, Tatoeba, Mewsli-X	morethan10
Javanese	Austronesian	Malayo-Polenisian	jv	both mBERT + XLM-R base/large	no monolingual models (but Javanese DistilBERT)		5	0,2	24	0	Latin/Javanese	none	NER, Tatoeba	NER, Tatoeba	lessthan100
Korean	Koreanic	Koreanic	ko	only mBERT	no monolingual models		9	54,2	5.644	185	Korean	none	PAWS-X, POS, NER, TyDiQA-GoldP, Tatoeba	POS, NER, TyDiQA-GoldP, Tatoeba	morethan10
Persian	Indo-European	Iranian	fa	only mBERT	all		9	111,6	13.259	6	Persian (derived from Arabic)	none	POS, NER, Tatoeba	POS, NER, Tatoeba, Mewsli-X	morethan10
Portuguese	Indo-European	Romance	pt	only mBERT	all		11	49,1	8.405	2.474	Latin	POS, WPR, QG	POS, NER, Tatoeba	POS, NER, Tatoeba	morethan10
Quechua	Quechuan	Quechuan	qu	both mBERT + XLM-R base/large	none	0,662	0	0	0	0	Latin	none	none	XCOPA, NER	lessthan100
Russian	Indo-European	East Slavic	ru	both mBERT + XLM-R base	all	0,453	12	278	23.408	904	Cyrillic	POS, NC, XNLI, NTG	XNLI, POS, NER, XQuAD, TyDiQA-GoldP, Tatoeba	XNLI, POS, NER, XQuAD, TyDiQA-GoldP, Tatoeba, LAReQA	morethan10
Spanish	Indo-European	Romance	es	both mBERT + XLM-R base	all	0,44	12	53,3	9.374	5.315	Latin	NER, POS, NC, MLQA, XNLI, PAWS-X, WPR, QG, NTG	XNLI, PAWS-X, POS, NER, XQuAD, MLQA, Tatoeba	XNLI, POS, NER, XQuAD, LAReQA, Mewsli-X	morethan10
Swahili	Atlantic-Congo	Bantoid	sw	both mBERT + XLM-R base/large	all	0,675	5	1,6	275	0	Latin/Arabic	none	XNLI, NER, TyDiQA-GoldP, Tatoeba	XNLI, POS, NER, TyDiQA-GoldP, Tatoeba, XCOPA	lessthan100
Turkish	Turkic	Oghuz	tr	both mBERT + XLM-R base	all	0,775	9	20,9	2.736	66	latin	POS, XNLI	XNLI, POS, NER, XQuAD, Tatoeba	XNLI, POS, NER, XQuAD, Tatoeba, LAReQA, Mewsli-X, XCOPA	morethan10
Vietnamese	Austroasiatic	Vietic	vi	both mBERT + XLM-R base	all	0,141	9	137,3	24.757	22	Latin	POS, XNLI	XNLI, POS, NER, XQuAD, MLQA, Tatoeba	XNLI, POS, NER, XQuAD, Tatoeba, LAReQA, XCOPA	morethan10

Figure B.1: Section of the table I used to aggregate important information about the individual languages, including the availability of language adapters and evaluation sets. Full table can be found on the thesis repository.

Bibliography

J. Ács. Exploring BERT's Vocabulary, 2019. URL http://juditacs.github.io/2019/02/19/bert-tokenization-stats.html.

D. I. Adelani, J. Abbott, G. Neubig, D. D'souza, J. Kreutzer, C. Lignos, C. Palen-Michel, H. Buzaaba, S. Rijhwani, S. Ruder, S. Mayhew, I. A. Azime, S. H. Muhammad, C. C. Emezue, J. Nakatumba-Nabende, P. Ogayo, A. Anuoluwapo, C. Gitau, D. Mbaye, J. Alabi, S. M. Yimam, T. R. Gwadabe, I. Ezeani, R. A. Niyongabo, J. Mukiibi, V. Otiende, I. Orife, D. David, S. Ngom, T. Adewumi, P. Rayson, M. Adeyemi, G. Muriuki, E. Anebi, C. Chukwuneke, N. Odu, E. P. Wairagala, S. Oyerinde, C. Siro, T. S. Bateesa, T. Oloyede, Y. Wambui, V. Akinode, D. Nabagereka, M. Katusiime, A. Awokoya, M. MBOUP, D. Gebreyohannes, H. Tilaye, K. Nwaike, D. Wolde, A. Faye, B. Sibanda, O. Ahia, B. F. P. Dossou, K. Ogueji, T. I. DIOP, A. Diallo, A. Akinfaderin, T. Marengereke, and S. Osei. MasakhaNER: Named entity recognition for African languages. *Transactions of the Association for Computational Linguistics*, 9:1116–1131, 2021. doi: 10.1162/tacl_a_00416. URL https://aclanthology.org/2021.tacl-1.66.

M. Artetxe and H. Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610, 2019. doi: 10.1162/tacl_a_00288. URL https://aclanthology.org/Q19-1038.

M. Artetxe, S. Ruder, and D. Yogatama. 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, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.421. URL https://aclanthology.org/2020.acl-main.421.

A. Bapna and O. Firat. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1165. URL https://aclanthology.org/D19-1165.

E. M. Bender. Linguistically naïve != language independent: Why NLP needs linguistic typology. In *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?*, pages 26–32, Athens, Greece, Mar. 2009. Association for Computational Linguistics. URL https://aclanthology.org/W09-0106.

E. M. Bender. On Achieving and Evaluating Language-Independence in NLP. *Linguistic Issues in Language Technology*, 6, Oct. 2011. ISSN 1945-3604. doi: 10.33011/lilt.v6i.1239. URL https://journals.colorado.edu/index.php/lilt/article/view/1239.

C. Bentz, T. Ruzsics, A. Koplenig, and T. Samardžić. A comparison between morphological complexity measures: Typological data vs. language corpora. In *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)*, pages 142–153, Osaka, Japan, Dec. 2016. The COLING 2016 Organizing Committee. URL https://aclanthology.org/W16-4117.

D. Bickerton. The language bioprogram hypothesis. *Behavioral and Brain Sciences*, 7 (2):173–188, 1984. doi: 10.1017/S0140525X00044149. Publisher: Cambridge University Press.

J. Bjerva and I. Augenstein. Does typological blinding impede cross-lingual sharing? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 480–486, Online, Apr. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.38. URL https://aclanthology.org/2021.eacl-main.38.

J. Bjerva, R. Östling, M. H. Veiga, J. Tiedemann, and I. Augenstein. What do language representations really represent? *Computational Linguistics*, 45(2):381–389, June 2019. doi: 10.1162/coli_a_00351. URL https://aclanthology.org/J19-2006.

C. Christodouloupoulos and M. Steedman. A massively parallel corpus: the Bible in 100 languages. *Language Resources and Evaluation*, 49(2):375–395, June 2015. ISSN 1574-0218. doi: 10.1007/s10579-014-9287-y. URL https://doi.org/10.1007/s10579-014-9287-y.

A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747.

R. Cotterell, S. J. Mielke, J. Eisner, and B. Roark. Are all languages equally hard to language-model? 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 536–541, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2085. URL https://aclanthology.org/N18-2085.

M. A. Covington and J. D. McFall. Cutting the Gordian Knot: The Moving-Average Type–Token Ratio (MATTR). *Journal of Quantitative Linguistics*, 17(2):94–100, May 2010. ISSN 0929-6174, 1744-5035. doi: 10.1080/09296171003643098. URL http://www.tandfonline.com/doi/abs/10.1080/09296171003643098.

W. de Vries, M. Wieling, and M. Nissim. Make the Best of Cross-lingual Transfer: Evidence from POS Tagging with over 100 Languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

Papers), pages 7676–7685, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.529. URL https://aclanthology.org/2022.acl-long.529.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

M. S. Dryer and M. Haspelmath, editors. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig, 2013. URL https://wals.info/.

X. Feng, X. Feng, L. Qin, B. Qin, and T. Liu. Language model as an annotator: Exploring DialoGPT for dialogue summarization. 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 1479–1491, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.117. URL https://aclanthology.org/2021.acl-long.117.

R. He, L. Liu, H. Ye, Q. Tan, B. Ding, L. Cheng, J. Low, L. Bing, and L. Si. On the effectiveness of adapter-based tuning for pretrained language model adaptation. 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 2208–2222, Online, Aug. 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.172. URL https://aclanthology.org/2021.acl-long.172.

X. He, Z. Cai, W. Wei, Y. Zhang, L. Mou, E. Xing, and P. Xie. Towards visual question answering on pathology images. 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 708–718, Online, Aug. 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.90. URL https://aclanthology.org/2021.acl-short.90.

N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-Efficient Transfer Learning for NLP. *Proceedings of the 36th International Conference on Machine Learning*, pages 2790–2799, 2019.

J. Hu, S. Ruder, A. Siddhant, G. Neubig, O. Firat, and M. Johnson. XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization. *arXiv:2003.11080 [cs]*, Sept. 2020. URL http://arxiv.org/abs/2003.11080. arXiv: 2003.11080.

A. Jones, W. Y. Wang, and K. Mahowald. A massively multilingual analysis of cross-linguality in shared embedding space. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5833–5847, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.471. URL https://aclanthology.org/2021.emnlp-main.471.

P. Joshi, S. Santy, A. Budhiraja, K. Bali, and M. Choudhury. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.560. URL https://aclanthology.org/2020.acl-main.560.

K. Kettunen. Can Type-Token Ratio be Used to Show Morphological Complexity of Languages? *Journal of Quantitative Linguistics*, 21(3):223–245, July 2014. ISSN 0929-6174, 1744-5035. doi: 10.1080/09296174.2014.911506. URL http://www.tandfonline.com/doi/abs/10.1080/09296174.2014.911506.

P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand, Sept. 13-15 2005. URL https://aclanthology.org/2005.mtsummit-papers.11.

X. L. Li and P. Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353.

Y. Liang, N. Duan, Y. Gong, N. Wu, F. Guo, W. Qi, M. Gong, L. Shou, D. Jiang, G. Cao, X. Fan, R. Zhang, R. Agrawal, E. Cui, S. Wei, T. Bharti, Y. Qiao, J.-H. Chen, W. Wu, S. Liu, F. Yang, D. Campos, R. Majumder, and M. Zhou. XGLUE: A new benchmark datasetfor cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.484. URL https://aclanthology.org/2020.emnlp-main.484.

Y.-H. Lin, C.-Y. Chen, J. Lee, Z. Li, Y. Zhang, M. Xia, S. Rijhwani, J. He, Z. Zhang, X. Ma, A. Anastasopoulos, P. Littell, and G. Neubig. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1301. URL https://aclanthology.org/P19-1301.

P. Littell, D. R. Mortensen, K. Lin, K. Kairis, C. Turner, and L. Levin. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain, Apr. 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-2002.

P. Liu, J. Fu, Y. Xiao, W. Yuan, S. Chang, J. Dai, Y. Liu, Z. Ye, and G. Neubig. ExplainaBoard: An explainable leaderboard for NLP. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 280–289, Online, Aug. 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-demo.34. URL https://aclanthology.org/2021.acl-demo.34.

P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *arXiv:2107.13586 [cs]*, July 2021b. URL http://arxiv.org/abs/2107.13586. arXiv: 2107.13586.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692*, July 2019. URL http://arxiv.org/abs/1907.11692. arXiv: 1907.11692.

R. K. Mahabadi, J. Henderson, and S. Ruder. Compacter: Efficient Low-Rank Hypercomplex Adapter Layers, Nov. 2021. URL http://arxiv.org/abs/2106.04647. Number: arXiv:2106.04647 arXiv:2106.04647 [cs].

T. Mayer and M. Cysouw. Creating a massively parallel Bible corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3158–3163, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/220_Paper.pdf.

S. J. Mielke, R. Cotterell, K. Gorman, B. Roark, and J. Eisner. What kind of language is hard to language-model? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4975–4989, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1491. URL https://aclanthology.org/P19-1491.

D. Moeljadi, F. Bond, and S. Song. Building an HPSG-based Indonesian resource grammar (INDRA). In *Proceedings of the Grammar Engineering Across Frameworks (GEAF) 2015 Workshop*, pages 9–16, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-3302. URL https://aclanthology.org/W15-3302.

D. Q. Nguyen and A. Tuan Nguyen. PhoBERT: Pre-trained language models for Vietnamese. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1037–1042, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.92. URL https://aclanthology.org/2020.findings-emnlp.92.

J. Nivre, M.-C. de Marneffe, F. Ginter, J. Hajič, C. D. Manning, S. Pyysalo, S. Schuster, F. Tyers, and D. Zeman. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.497.

H. H. Park, K. J. Zhang, C. Haley, K. Steimel, H. Liu, and L. Schwartz. Morphology matters: A multilingual language modeling analysis. *Transactions of the Association for Computational Linguistics*, 9:261–276, 2021. doi: 10.1162/tacl_a_00365. URL https://aclanthology.org/2021.tacl-1.16.

J. Pfeiffer, A. Rücklé, C. Poth, A. Kamath, I. Vulić, S. Ruder, K. Cho, and I. Gurevych. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020*

Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 46–54, Online, Oct. 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.7. URL https://aclanthology.org/2020.emnlp-demos.7.

J. Pfeiffer, I. Vulić, I. Gurevych, and S. Ruder. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online, Nov. 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.617. URL https://aclanthology.org/2020.emnlp-main.617.

J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho, and I. Gurevych. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online, Apr. 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.39. URL https://aclanthology.org/2021.eacl-main.39.

J. Pfeiffer, I. Vulić, I. Gurevych, and S. Ruder. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic, Nov. 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.800. URL https://aclanthology.org/2021.emnlp-main.800.

J. Phang, I. Calixto, P. M. Htut, Y. Pruksachatkun, H. Liu, C. Vania, K. Kann, and S. R. Bowman. English intermediate-task training improves zero-shot cross-lingual transfer too. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 557–575, Suzhou, China, Dec. 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.aacl-main.56.

T. Pires, E. Schlinger, and D. Garrette. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1493. URL https://aclanthology.org/P19-1493.

E. M. Ponti, H. O'Horan, Y. Berzak, I. Vulić, R. Reichart, T. Poibeau, E. Shutova, and A. Korhonen. Modeling language variation and universals: A survey on typological linguistics for natural language processing. *Computational Linguistics*, 45(3):559–601, Sept. 2019. doi: 10.1162/coli_a_00357. URL https://aclanthology.org/J19-3005.

E. M. Ponti, G. Glavaš, O. Majewska, Q. Liu, I. Vulić, and A. Korhonen. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.185. URL https://aclanthology.org/2020.emnlp-main.185.

S. Ruder, N. Constant, J. Botha, A. Siddhant, O. Firat, J. Fu, P. Liu, J. Hu, D. Garrette, G. Neubig, and M. Johnson. XTREME-R: Towards more challenging and nuanced multilingual evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10215–10245, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.802. URL https://aclanthology.org/2021.emnlp-main.802.

P. Rust, J. Pfeiffer, I. Vulić, S. Ruder, and I. Gurevych. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.243. URL https://aclanthology.org/2021.acl-long.243.

M. Schuster and K. Nakajima. Japanese and Korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152, Kyoto, Japan, Mar. 2012. IEEE. ISBN 978-1-4673-0046-9 978-1-4673-0045-2 978-1-4673-0044-5. doi: 10.1109/ICASSP.2012.6289079. URL http://ieeexplore.ieee.org/document/6289079/.

M. Straka, J. Hajič, and J. Straková. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://aclanthology.org/L16-1680.

J. Tiedemann. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf.

J. Tiedemann. The tatoeba translation challenge – realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online, Nov. 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.wmt-1.139.

University of Helsinki, FIN-CLARIN, J. Rueter, and E. Axelson. Parallel Bible Verses for Uralic Studies, Korp. text corpus, Kielipankki, 2020. URL http://urn.fi/urn:nbn:fi:lb-2020021121.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin. Attention is All you Need. *31st Conference on Neural Information Processing Systems (NIPS 2017)*, page 11, 2017.

S. Wu and M. Dredze. 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, Nov.

2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1077. URL https://aclanthology.org/D19-1077.

S. Wu and M. Dredze. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.repl4nlp-1.16. URL https://aclanthology.org/2020.repl4nlp-1.16.

I. Zeroual and A. Lakhouaja. MulTed: a multilingual aligned and tagged parallel corpus. *Applied Computing and Informatics*, 18(1/2):61–73, Jan. 2022. ISSN 2210-8327, 2210-8327. doi: 10.1016/j.aci.2018.12.003. URL https://www.emerald.com/insight/content/doi/10.1016/j.aci.2018.12.003/full/html.

P. Zweigenbaum, S. Sharoff, and R. Rapp. Overview of the Third BUCC Shared Task: Spotting Parallel Sentences in Comparable Corpora. *Proceedings of the 11th Workshop on Building and Using Comparable Corpora at LREC 2018*, page 4, 2018.