HOW DO MULTILINGUAL ENCODERS LEARN CROSS-LINGUAL REPRESENTATION?

by

Shijie Wu

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

November, 2021

Abstract

NLP systems typically require support for more than one language. As different languages have different amounts of supervision, cross-lingual transfer benefits languages with little to no training data by transferring from other languages. From an engineering perspective, multilingual NLP benefits development and maintenance by serving multiple languages with a single system. Both cross-lingual transfer and multilingual NLP rely on cross-lingual representations serving as the foundation. As BERT revolutionized representation learning and NLP, it also revolutionized cross-lingual representations and cross-lingual transfer. Multilingual BERT was released as a replacement for single-language BERT, trained with Wikipedia data in 104 languages.

Surprisingly, without any explicit cross-lingual signal, multilingual BERT learns cross-lingual representations in addition to representations for individual languages. This thesis first shows such surprising cross-lingual effectiveness compared against prior art on various tasks. Naturally, it raises a set of questions, most notably how do these multilingual encoders learn cross-lingual representations. In exploring these questions, this thesis will analyze the behavior of multilingual models in a variety of settings on high and low resource languages.

ABSTRACT

We also look at how to inject different cross-lingual signals into multilingual encoders, and the optimization behavior of cross-lingual transfer with these models. Together, they provide a better understanding of multilingual encoders on cross-lingual transfer. Our findings will lead us to suggested improvements to multilingual encoders and cross-lingual transfer.

Readers: Mark Dredze, Benjamin Van Durme, João Sedoc

Acknowledgments

This thesis is the product of 39 months of work. In this journey, over 20 months were spent in my home without any in-person meeting with my advisor and labmates, due to the global pandemic known as COVID-19. It was not the easiest time. I owe gratitude to many people who helped me throughout my Ph.D. journey, as well as many more who made that journey possible. I would try my best to thank everyone who helped me along this journey.

I am tremendously thankful to my adviser, Mark Dredze. Mark recruited me to stay at Hopkins and has supported me throughout my Ph.D. journey. Mark offered me the freedom to dive into the research of multilingual encoders and guided me on both research and life as a grad student.

I would also like to thank Benjamin Van Durme and João Sedoc for serving on my committee and for mentoring me during the IARPA BETTER program.[1].

Before my Ph.D. journey, I spent two years at Hopkins as a Master student. I am thankful to Jason Eisner, who taught the NLP course, sparked my interest in NLP, and offered me an opportunity in NLP research. I am also thankful to Ryan Cotterell, who has collaborated

[1] 2019-19051600005

with me on several papers.

No man is an island. I would like to thank my other co-authors: Aaron, Alex, Mans, Alexis, Chaitanya, Craig, Edo, Guanghui, Haoran, Haoran, Jialiang, Kenton, Luke, Mahsa, Marc, Micha, Nizar, Pamela, Patrick, Seth, Tim, Ves, and Yunmo. I would also like to thank everyone who contributed to the SIGMORPHON shared tasks in 2019, 2020, and 2021. While some of the works are not part of this thesis, this thesis will use plural pronouns after these acknowledgments to reflect that this work is not all mine.

My most heartfelt thanks goes to my family. My parents supported my curiosity in science and my pursuit of education. They gave me the freedom and support to pursue my interests, from taking me to the library to letting me pick any extracurricular classes. I could not enumerate how much they have helped me throughout my education journey. I would also like to thank my dog Lucky, who brings me joy in the final month of this journey during the writing of this thesis. My final thanks goes to Gege, who has shown me love and support every day in the past seven years.

Dedication

As Max Weber put it in his speech "Science as a Vocation",

> In science, each of us knows that what he has accomplished will be antiquated in ten, twenty, fifty years. That is the fate to which science is subjected; it is the very meaning of scientific work, to which it is devoted in a quite specific sense, as compared with other spheres of culture for which in general the same holds. Every scientific 'fulfilment' raises new 'questions'; it asks to be 'surpassed' and outdated. Whoever wishes to serve science has to resign himself to this fact. Scientific works certainly can last as 'gratifications' because of their artistic quality, or they may remain important as a means of training. Yet they will be surpassed scientifically–let that be repeated–for it is our common fate and, more, our common goal. We cannot work without hoping that others will advance further than we have. In principle, this progress goes on ad infinitum.

I dreamed to become a scientist when I was a kid, but I did not understand what it meant until I read this paragraph. NLP, ML, and AI change much faster today than science did a hundred years ago. What you and I accomplished would be antiquated in less than two or even one year, so that together as a community, we could keep moving forward faster and faster. Therefore, I am dedicating this thesis to the ad infinitum of scientific progress, answering open questions and pushing the frontier forward. I hope that I get to witness the modeling of human language on the same level or even surpass our brain one day.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Modern NLP applications typically require support for more than one language, and we want to build an equally good system for each language. As different languages usually do not have the same amount of supervision, for languages with the least or even no training data, we may rely on cross-lingual transfer—transferring knowledge from languages with more supervision to languages with less or even no supervision. From an engineering perspective, managing different systems for different languages introduces challenges for continuous development and maintenance. Thus, even if we had enough training data for each language to build one system per language, therefore eliminating the need for cross-lingual transfer, we may still want a single system for all languages, namely a multilingual system.

Cross-lingual transfer and multilingual NLP both greatly benefit from cross-lingual representation. Supposed we have access to perfect cross-lingual representation space, i.e. words with similar meaning across languages have similar vector representation, transferring knowledge across language would be straight-forward. Similarly, with such representation, multilingual NLP models only need to learn to solve the task without worrying how to encode words into vectors, leaving much less to learn. Looking at the literature overall, the quality of cross-lingual representation tends to improve as representation learning techniques improve. NLP has moved from hand-engineered features with classical machine learning models to word embeddings with deep neural networks. In the past four years, representation learning methods like ELMo—a deep LSTM network trained with language model objective—and BERT—a deep Transformer network trained with masked language model objective—have

revolutionized NLP again, including cross-lingual representation learning. In this thesis, we will refer to models like ELMo and BERT as encoders, encoding words in context into contextual vector representation with deep networks.

Around three years ago, in November 2018, a multilingual version of BERT was released, called multilingual BERT (mBERT). As the authors of BERT say "[...] (they) do not plan to release more single-language models", they instead train a single BERT model with Wikipedia to serve 104 languages, hence multilingual BERT. The main difference between English monolingual BERT and multilingual BERT is the training data: Wikipedia of English v.s. Wikipedia of 104 languages. Surprisingly, even without any explicit cross-lingual signal during pretraining, mBERT shows promising zero-shot cross-lingual performance—training the model on one language then directly applying that model to another language—on a natural language inference dataset.

This thesis first fully documents the surprising cross-lingual potential of mBERT on various tasks against prior art via zero-shot cross-lingual transfer, which directly tests its cross-lingual representation. We show that mBERT is not only learning representation for each language but also learning cross-lingual representation. Such surprising cross-lingual effectiveness leads to a set of questions, most importantly how do multilingual encoders learn cross-lingual representations. This thesis attempts to answer these questions, in doing so, to better understand models behaviour and how these models learn cross-lingual representation. With these insights, we are able to identify and improve their cross-lingual representation and cross-lingual transfer with these encoders.

1.1 Outline & Contributions

The main contribution of this thesis is to understand how multilingual encoders learn cross-lingual representations. In exploring this question, we will analyse the behavior of multilingual models in a variety of settings on high and low resource languages. Our findings will lead us to suggested improvements to these models, the testing of which will allow us to better understand how these models work and what makes them effective. As we document the surprising cross-lingual effectiveness of these multilingual models, in each chapter, we answer different questions raised by these models. Together, they provide a better understanding of multilingual encoders on cross-lingual transfer, which leads to directions to improve these models for cross-lingual transfer. All chapters are supported by the same codebase `https://github.com/shijie-wu/crosslingual-nlp`.

Chapter 2 reviews the progress of representation learning in NLP and discusses its application in cross-lingual transfer. Improvement on representation learning typically leads to better cross-lingual representation. As BERT revolutionizes representation learning and NLP, a multilingual version of BERT called Multilingual BERT (mBERT) is also released.

Does mBERT learn cross-lingual representation? In chapter 3, we show that surprisingly mBERT learns cross-lingual representation even without explicit cross-lingual signal, even outperforming previous state-of-the-art cross-lingual word embeddings on zero-shot cross-lingual transfer. Additionally, we probe mBERT and document the model behavior. This work was published in Wu and Dredze (2019).

How does mBERT learn cross-lingual representation? Chapter 4 presents an ablation

study on mBERT, teasing apart which modeling decision contributes the most to the learning of cross-lingual representation. We show that sharing transformer parameters is the most important factor. As monolingual BERT of different languages are similar to each other, parameter sharing allows the model to natrually align the representation in a cross-lingual fashion. This work was published in Conneau et al. (2020b).

Are all languages created equal in mBERT? In chapter 5, we show that mBERT does not learn equally high quality representation for its lower resource languages. Such outcome is not the product of hyperparameter or multilingual joint training but the sample inefficiency of BERT objective, as monolingual BERT of these languages perform even worse and pairing them with similar high resource languages close the performance gap. This work was published in Wu and Dredze (2020a).

How to inject cross-lingual signals into multilingual encoders? Chapter 6 introduces two approaches for injecting two types of cross-lingual signal into multilingual encoders: bilingual dictionary and bitext. For the former, we create synthetic code-switch corpus for pretraining. For the latter, we ad-hoc explicitly align the encoder representation using a contrastive alignment loss. Both methods show improvement for mBERT or smaller encoders. However, the performance gain is eclipsed by simply scaling up the model size and data size. Additionally, we observe that zero-shot cross-lingual transfer has high variance on the target language, creating challenges for comparing models fairly in the literature. This work was published in Conneau et al. (2020b) and Wu and Dredze (2020b).

Why does zero-shot cross-lingual transfer have high variance as shown in chapter 6?

In chapter 7, we show that zero-shot cross-lingual transfer is under-specified optimization, causing its high variance on target languages and much lower variance on source language. To improve the performance of zero-shot cross-lingual transfer, addressing the under-specification could produce bigger gain.

Does data projection constrain zero-shot cross-lingual transfer optimization? Chapter 8 proposes using silver target data—created automatically with machine translation based on supervision in source language—to constrain the optimization, and shows adding such constraint improves zero-shot cross-lingual transfer. We also investigate the impact of encoder on the data creation pipeline, and observe that the best setup is task specific. This work was published in Yarmohammadi et al. (2021).

Chapter 9 recaps our contributions and discusses future work.

Chapter 2

Representation Learning in NLP

In the past decade, representation learning has improved natural language processing (NLP) technology significantly. Representation learning learns dense representation of language using unlabeled corpus, using the corpus itself as a learning signal. As the computational infrastructure scales with the collection of Web corpus, the capability of representation learning keeps scaling (Radford et al., 2019). Every sub-fields within NLP has been revolutionized by representation learning, including cross-lingual transfer. Cross-lingual transfer attempts to transfer knowledge from one language—typically languages with lots of supervision—to another language—typically languages with less supervision. As modern NLP technology is deployed to support more than one language and different languages have different amounts of supervision for tasks of interest, cross-lingual transfer is the bedrock of NLP real world application. In this chapter, we will discuss the progress on representation learning in NLP and its impact on cross-lingual transfer.

2.1 Word Embeddings

Word embeddings encode word to dense vector representation. It had existed as a part of the neural network based NLP model before, such as language model (Bengio et al., 2003). While global matrix factorization based methods for learning word embeddings have existed for decades, such as latent semantic analysis (Deerwester et al., 1990) and Brown clusters (Brown et al., 1992), online learning based approaches like Word2Vec (skip-gram and CBOW) (Mikolov et al., 2013a; Mikolov et al., 2013b), Glove (Pennington, Socher,

and Manning, 2014), and FastText (Bojanowski et al., 2017) pushes representation learning to a prominent position in NLP. Pretrained word embeddings became a standalone step in the pipeline of developing neural NLP systems with it as input to the neural network. **Pretraining** usually refers to the training procedure of word embeddings and later contextual word embeddings, as it is learning information from the corpus itself instead of from any particular tasks.

The learning of embedding of a word relies on its contextual information, as the distributional hypothesis states that words in similar contexts have similar meanings. Specifically, skip-gram trains a log-bilinear model to predict words within a certain window size using only the center word, while CBOW trains a similar model to predict the center word using a bag of context words. Both skip-gram and CBOW approximate the word prediction softmax loss with noise contrastive estimation and negative sampling. Glove instead trains word embeddings to predict global co-occurrence word statistics. FastText additionally extends Word2Vec by incorporating subword information. Due to efficiency consideration, word embeddings represent each word type with a single fixed-dimensional vector, trained with local co-occurrence signals regardless of order. As deep learning framework and computational infrastructure improves, such limitations would be later addressed by contextual word embeddings.

2.2 Contextual Word Embeddings

Different from word embeddings, contextual word embeddings represents word using its context processed by a deep neural network. There are many attempts on such idea with language model (Peters et al., 2017) or machine translation (McCann et al., 2017) as learning signal, and ELMo (Peters et al., 2018) popularized it within the NLP community. ELMo, two deep LSTM (Hochreiter and Schmidhuber, 1997) pretrained with right-to-left and left-to-right language modeling objective, produce contextual word embeddings by combining the output of each layer of LSTM with weighted averaging. Additionally, convolution is used to encode character-level information. This contextualized representation outperforms stand-alone word embeddings, e.g. Word2Vec and Glove, with the same task-specific architecture in various downstream tasks, and achieves state-of-the-art performance at the time of publication. Similar to word embeddings, neural network takes static representation from ELMo as input.

Instead of taking the representation from a pretrained model, GPT (Radford et al., 2018) and Howard and Ruder (2018) also fine-tune all the parameters of the pretrained model for a specific task, referred to as **fine-tuning**. Also, GPT uses a transformer encoder (Vaswani et al., 2017) instead of an LSTM and jointly fine-tunes with the language modeling objective. Howard and Ruder (2018) propose another fine-tuning strategy by using a different learning rate for each layer with learning rate warmup and gradual unfreezing.

2.3 BERT

BERT (Devlin et al., 2019) is a deep contextual representation based on a series of transformers trained by a self-supervised objective. One of the main differences between BERT and related work like ELMo and GPT is that BERT is trained by the Cloze task (Taylor, 1953), also referred to as masked language modeling, instead of right-to-left or left-to-right language modeling. This allows the model to freely encode information from both directions in each layer, contributing to its better performance compared to ELMo and GPT. The goal of the Cloze task is to predict the center missing word based on its context. BERT could be viewed as a deep CBOW, using much deeper representation to encode much larger ordered contextual information. Softmax is used to compute the probability of the missing word based on the contextual representation.

To set up the Cloze for training, the authors propose a heuristic to replace each word with a mask or a random word with the probability of 12% or 1.5%, respectively. Additionally, BERT also optimizes a next sentence classification objective. At training time, 50% of the paired sentences are consecutive sentences while the rest of the sentences are paired randomly. Instead of operating on words, BERT uses a subword vocabulary with WordPiece (Wu et al., 2016), a data-driven approach to break up a word into subwords. Using a subword vocabulary allows BERT to keep a modest vocabulary size, making the softmax prediction practical, and offers a balance between word-based vocabulary and character-level encoding like ELMo.

2.3.1 Fine-tuning

BERT shows state-of-the-art performance at the time of publication by fine-tuning the transformer encoder followed by a simple softmax classification layer on sentence classification tasks, and a sequence of shared softmax classifications for sequence tagging models on tasks like NER. Fine-tuning usually takes 3 to 4 epochs with a relatively small learning rate, for example, 3e-5. Instead of directly fine-tuning BERT on the task of interests, Phang, Févry, and Bowman (2018) propose intermediate fine-tuning—fine-tuning BERT on data-rich supervised tasks—and show improvement on the final task of interests. Unlike the later GPT-2 and GPT-3, which will be discussed in Section 2.5, BERT typically requires task-specific data fine-tuning to perform said tasks.

2.4 Transformer

Since the introduction of transformer, it has taken over NLP. Its popularity could be attributed to two factors: easy to parallelize and easy to model long range context. For a sequence with length n, while recurrent-based models like RNN and LSTM have $O(n)$ sequential operation, transformer has $O(1)$ sequential operation in comparison, making it much more parallelizable. Additionally, to connect any two items within a sequence, recurrent-based models need to pass through up to $O(n)$ items in between while transformer directly connect these two items, making it much easier to model long context.

For completeness, we describe the Transformer used by BERT. Let $\mathbf{x}, \mathbf{y}$ be a sequence

of subwords from a sentence pair. A special token `[CLS]` is prepended to $\mathbf{x}$ and `[SEP]` is appended to both $\mathbf{x}$ and $\mathbf{y}$. The embedding is obtained by

$$\hat{h}_i^0 = E(x_i) + E(i) + E(\mathbb{1}_{\mathbf{x}}) \tag{2.1}$$

$$\hat{h}_{j+|\mathbf{x}|}^0 = E(y_j) + E(j + |\mathbf{x}|) + E(\mathbb{1}_{\mathbf{y}}) \tag{2.2}$$

$$h_\cdot^0 = \text{Dropout}(\text{LN}(\hat{h}_\cdot^0)) \tag{2.3}$$

where E is the embedding function and LN is layer normalization (Ba, Kiros, and Hinton, 2016). M transformer blocks are followed by the embeddings. In each transformer block,

$$h_\cdot^{i+1} = \text{Skip}(\text{FF}, \text{Skip}(\text{MHSA}, h_\cdot^i)) \tag{2.4}$$

$$\text{Skip}(f, h) = \text{LN}(h + \text{Dropout}(f(h))) \tag{2.5}$$

$$\text{FF}(h) = \text{GELU}(h\mathbf{W}_1^\top + \mathbf{b}_1)\mathbf{W}_2^\top + \mathbf{b}_2 \tag{2.6}$$

where GELU is an element-wise activation function (Hendrycks and Gimpel, 2016). In practice, $h^i \in \mathbb{R}^{(|\mathbf{x}|+|\mathbf{y}|)\times d_h}$, $\mathbf{W}_1 \in \mathbb{R}^{4d_h \times d_h}$, $\mathbf{b}_1 \in \mathbb{R}^{4d_h}$, $\mathbf{W}_2 \in \mathbb{R}^{d_h \times 4d_h}$, and $\mathbf{b}_2 \in \mathbb{R}^{d_h}$. MHSA is the multi-heads self-attention function. We show how one new position $\hat{h}_i$ is computed.

$$[\cdots, \hat{h}_i, \cdots] = \text{MHSA}([h_1, \cdots, h_{|\mathbf{x}|+|\mathbf{y}|}]) \tag{2.7}$$

$$= \mathbf{W}_o \text{Concat}(h_i^1, \cdots, h_i^N) + \mathbf{b}_o \tag{2.8}$$

In each attention, referred to as attention head,

$$h_i^j = \sum_{k=1}^{|\mathbf{x}|+|\mathbf{y}|} \text{Dropout}(\alpha_k^{(i,j)}) \mathbf{W}_V^j h_k \tag{2.9}$$

$$\alpha_k^{(i,j)} = \frac{\exp \frac{(\mathbf{W}_Q^j h_i)^\top \mathbf{W}_K^j h_k}{\sqrt{d_h/N}}}{\sum_{k'=1}^{|\mathbf{x}|+|\mathbf{y}|} \exp \frac{(\mathbf{W}_Q^j h_i)^\top \mathbf{W}_K^j h_{k'}}{\sqrt{d_h/N}}} \tag{2.10}$$

where N is the number of attention heads, $h_i^j \in \mathbb{R}^{d_h/N}$, $\mathbf{W}_o \in \mathbb{R}^{d_h \times d_h}$, $\mathbf{b}_o \in \mathbb{R}^{d_h}$, and $\mathbf{W}_Q^j, \mathbf{W}_K^j, \mathbf{W}_V^j \in \mathbb{R}^{d_h/N \times d_h}$.

2.5 Generative Language Model

Generative language model (LM) pretrained with language modeling objective. In this sense, ELMo and GPT are both generative LM. However, generative LM typically also refer to how the model was used. Instead of taking the representation from the model like ELMo or GPT, they instead cast the task of interest as language modeling, e.g. GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020). Any natural language generation tasks fall under this category, and some NLP tasks can be naturally cast as language model with prompt and template. GPT-2 shows pretrained generative LM can performs zero-shot learning on some tasks—no fine-tuning is needed. GPT-3 shows larger generative LM can performs few-shot learning with only context—again no fine-tuning is needed—although typical few-shot learning usually involve fine-tuning. One potential reason for no fine-tuning in GPT-3 could be model size with 175B parameters, making fine-tuning expensive. This thesis does not

focus on generative LM as there is no publicly available multilingual generative LM until the recent mT5 (Xue et al., 2021), a model with both encoder like BERT and decoder like GPT pretrained with span-corruption objective (Raffel et al., 2020). However, scaling beyond the current biggest encoder might need generative LM, as we discuss in chapter 9.

2.6 Cross-lingual Transfer and Multilingual NLP

Cross-lingual transfer learning is a type of transductive transfer learning with different source and target domain (Pan and Yang, 2010). It attempts to transfer knowledge from one language, usually referred to as source language, to another language, usually referred to as target language. It is possible to have more than one source language or more than one target language. **Few-shot cross-lingual transfer** assumes limited training data in target languages, while **Zero-shot cross-lingual transfer** typically assumes no task specific supervision on target language. A stricter assumption further eliminates any cross-lingual signal like bilingual dictionary or bitext. A cross-lingual representation space is assumed to perform the cross-lingual transfer, and the quality of the cross-lingual space is essential for cross-lingual transfer, especially zero-shot transfer. Multilingual NLP attempts to build a single NLP system supporting multiple languages. Cross-lingual representation benefits the development of multilingual NLP, as it alleviates the learning to solve the specific task. This thesis mainly focuses on zero-shot cross-lingual transfer as a proxy for evaluating cross-lingual representation.

2.7 Cross-lingual Representation

Cross-lingual representation learning follows a similar development trajectory as representation learning in NLP. Before the widespread use of cross-lingual word embeddings, task-specific models assumed coarse-grain representation like part-of-speech tags, in support of a delexicalized parser (Zeman and Resnik, 2008).

2.7.1 Cross-lingual Word Embeddings

With the progress on word embeddings, Mikolov, Le, and Sutskever (2013) shows that embedding spaces tend to be shaped similarly across different languages. This inspired work in aligning monolingual embeddings. The alignment was done by using a bilingual dictionary to project words that have the same meaning close to each other with linear mapping (Mikolov, Le, and Sutskever, 2013). This projection aligns the words outside of the dictionary as well due to the similar shapes of the word embedding spaces. Follow-up efforts only required a very small seed dictionary (e.g., only numbers (Artetxe, Labaka, and Agirre, 2017)) or even no dictionary at all (Lample et al., 2018; Zhang et al., 2017). Ruder, Vulić, and Søgaard (2019) surveys methods for learning cross-lingual word embeddings by either joint training or post-training mappings of monolingual embeddings. Other work has pointed out that word embeddings may not be as isomorphic as thought (Søgaard, Ruder, and Vulić, 2018) especially for distantly related language pairs (Patra et al., 2019). Ormazabal et al. (2019) show joint training can lead to more isomorphic word embeddings space. On

top of cross-lingual word embeddings, task-specific neural architectures have been used for tasks like named entity recognition (Xie et al., 2018), part-of-speech tagging (Kim et al., 2017) and dependency parsing (Ahmad et al., 2019).

2.7.2 Cross-lingual Contextual Word Embeddings

However, cross-lingual word embeddings have similar drawbacks as word embeddings. With the success of ELMo over word embeddings, Schuster et al. (2019) aligns pretrained ELMo of different languages by learning an orthogonal mapping and shows strong zero-shot and few-shot cross-lingual transfer performance on dependency parsing with 5 Indo-European languages. Mulcaire, Kasai, and Smith (2019) trains a single ELMo on distantly related languages and shows mixed results as to the benefit of pretraining.

2.8 Multilingual BERT

BERT offers a multilingual model, called Multilingual BERT (mBERT), pretrained on concatenated Wikipedia data for 104 languages *without any explicit cross-lingual signal*, e.g. pairs of words, sentences or documents linked across languages (Devlin, 2018). It follows the same model architecture and training procedure as BERT, except with data from Wikipedia in 104 languages.

In mBERT, the WordPiece modeling strategy allows the model to share embeddings across languages. For example, “DNA” has a similar meaning even in distantly related

languages like English and Chinese.[1] To account for varying sizes of Wikipedia training data in different languages, training uses a heuristic to subsample or oversample words when running WordPiece as well as sampling a training batch, random words for cloze and random sentences for next sentence classification.

However, mBERT does surprisingly well compared to cross-lingual word embeddings on zero-shot cross-lingual transfer in XNLI (Conneau et al., 2018), a natural language inference dataset (Devlin, 2018). While the XNLI experiment is promising, many questions remain unanswered. What does mBERT really learn: separate representation for each language, or some cross-lingual representation mixed with some language-specific representation? Is mBERT better than cross-lingual word embeddings in terms of cross-lingual transfer? How does its modeling decision impact its performance? In chapter 3, we will conduct experiments to answer these questions.

[1] "DNA" indeed appears in the vocabulary of mBERT as a stand-alone lexicon.

Chapter 3

Does mBERT Learn Cross-lingual Representation?

3.1 Introduction

As we discuss in Section 2.8, while XNLI results are promising, the question remains: does mBERT learn a cross-lingual space that supports zero-shot transfer, even without any explicit cross-lingual signal? Does mBERT learn a cross-lingual representation, or does it produce a representation for each language in its own embedding space? Is mBERT better than cross-lingual word embeddings in terms of cross-lingual transfer? How does its modeling decision impact its performance?

In this chapter, grounded by models in the literature, we evaluate mBERT as a zero-shot cross-lingual transfer model on five different NLP tasks: natural language inference, document classification, named entity recognition, part-of-speech tagging, and dependency parsing. We show that it achieves competitive or even state-of-the-art performance (at the time of publication) by simply fine-tuning all parameter of mBERT with minimal task-specific layer. This is surprising as mBERT does not have any explicit cross-lingual signal during pretraining while prior work assume various amount of cross-lingual signal. While fine-tuning all parameters achieves strong performance, we additionally explore different fine-tuning and feature extraction schemes. We demonstrate that we could further outperform the suggested fine-tune all approach with simple parameter freezing—freezing the bottom layer of mBERT. Furthermore, we explore the extent to which mBERT maintains language-specific information by probing each layer of mBERT with language identification. Surprisingly, mBERT maintains strong language specific information despite having strong cross-lingual representation. Finally, we show how subword tokenization modeling decisions

	ar	bg	ca	cs	da	de	el	en	es	et	fa	fi	fr	he	hi	hr	hu	id	it	ja	ko	la	lv	nl	no	pl	pt	ro	ru	sk	sl	sv	sw	th	tr	uk	ur	vi	zh
MLDoc						✓		✓	✓				✓						✓	✓									✓											✓
NLI	✓	✓				✓	✓	✓	✓				✓		✓														✓				✓	✓	✓		✓	✓	✓
NER						✓		✓	✓															✓															✓
POS		✓			✓	✓		✓	✓		✓						✓		✓					✓		✓	✓	✓		✓	✓	✓						
Parsing	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				✓			✓

Table 3.1: The 39 languages used in the 5 tasks.

impact cross-lingual transfer performance. We observe a positive correlation between the amount of subword overlap between languages and the transfer performance across languages.

Parallel to the publication of this chapter, Conneau and Lample (2019) incorporates bitext into BERT by training on pairs of parallel sentences. Pires, Schlinger, and Garrette (2019) shows mBERT has good zero-shot cross-lingual transfer performance on NER and POS tagging. They show how subword overlap and word ordering affect mBERT transfer performance. Additionally, they show mBERT can find translation pairs and works on code-switched POS tagging. In comparison, this chapter looks at a larger set of NLP tasks including dependency parsing and ground the mBERT performance against previous state-of-the-art on zero-shot cross-lingual transfer. We also probe mBERT in different ways and show a more complete picture of the cross-lingual effectiveness of mBERT.

3.2 Tasks

We consider five tasks in the zero-shot transfer setting. We assume labeled training data for each task in English, and transfer the trained model to a target language. We select a range of different tasks: document classification, natural language inference, named entity

recognition, part-of-speech tagging, and dependency parsing. We cover zero-shot transfer from English to 38 languages in the 5 different tasks as shown in Table 3.1. In this section, we describe the tasks as well as task-specific layers.

3.2.1 Document Classification

We use MLDoc (Schwenk and Li, 2018), a balanced subset of the Reuters corpus covering 8 languages for document classification. The 4-way topic classification task decides between CCAT (Corporate/Industrial), ECAT (Economics), GCAT (Government/Social), and MCAT (Markets). We only use the first two sentences[1] of a document for classification due to memory constraint. The sentence pairs are provided to the mBERT encoder. The task-specific classification layer is a linear function mapping $h_0^{12} \in \mathbb{R}_h^d$ into $\mathbb{R}^4$, and a softmax is used to get class distribution. We evaluate by classification accuracy.

3.2.2 Natural Language Inference

We use XNLI (Conneau et al., 2018) which covers 15 languages for natural language inference. The 3-way classification includes entailment, neutral, and contradiction given a pair of sentences. We feed a pair of sentences directly into mBERT and the task-specific classification layer is the same as Section 3.2.1. We evaluate by classification accuracy.

[1]We only use the first sentence if the document only contains one sentence. Documents are segmented into sentences with NLTK (Perkins, 2014).

3.2.3 Named Entity Recognition

We use the CoNLL 2002 and 2003 NER shared tasks (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003) (4 languages) and a Chinese NER dataset (Levow, 2006). The labeling scheme is BIO with 4 types of named entities. We add a linear classification layer with softmax to obtain word-level predictions. Since mBERT operates at the subword-level while the labeling is word-level, if a word is broken into multiple subwords, we mask the prediction of non-first subwords. NER is evaluated by F1 of predicted entities (F1). Note we adopt a simple post-processing heuristic to obtain a valid span, rewriting standalone `I-X` into `B-X` and `B-X I-Y I-Z` into `B-Z I-Z I-Z`, following the final entity type.

3.2.4 Part-of-Speech Tagging

We use a subset of Universal Dependencies (UD) Treebanks (v1.4) (Nivre, 2016), which cover 15 languages, following the setup of Kim et al. (2017). The task-specific labeling layer is the same as Section 3.2.3. POS tagging is evaluated by the accuracy of predicted POS tags (ACC).

3.2.5 Dependency parsing

Following the setup of Ahmad et al. (2019), we use a subset of Universal Dependencies (UD) Treebanks (v2.2) (Nivre, 2018a), which includes 31 languages. Dependency parsing is

evaluated by unlabelled attachment score (UAS) and labeled attachment score (LAS) [2]. We only predict the coarse-grain dependency label following Ahmad et al. We use the model of Dozat and Manning (2017), a graph-based parser as a task-specific layer. Their LSTM encoder is replaced by mBERT. Similar to Section 3.2.3, we only take the representation of the first subword of each word. We use masking to prevent the parser from operating on non-first subwords.

3.3 Experiments

We use the base cased multilingual BERT, which has $N = 12$ attention heads and $M = 12$ transformer blocks. The dropout probability is 0.1 and d_h is 768. The model has 179M parameters with about 120k vocabulary.

3.3.1 Training

For each task, no preprocessing is performed except tokenization of words into subwords with WordPiece. We use Adam (Kingma and Ba, 2014) for fine-tuning with β_1 of 0.9, β_2 of 0.999 and L2 weight decay of 0.01. We warm up the learning rate over the first 10% of batches and linearly decay the learning rate.

[2]Punctuations (PUNCT) and symbols (SYM) are excluded.

3.3.2 Maximum Subwords Sequence Length

At training time, we limit the length of subwords sequence to 128 to fit in a single GPU for all tasks. For NER and POS tagging, we additionally use the sliding window approach. After the first window, we keep the last 64 subwords from the previous window as context. In other words, for a non-first window, only (up to) 64 new subwords are added for prediction. At evaluation time, we follow the same approach as training time except for parsing. We threshold the sentence length to 140 words, including words and punctuation, following Ahmad et al. (2019). In practice, the maximum subwords sequence length is the number of subwords of the first 140 words or 512, whichever is smaller.

3.3.3 Hyperparameter Search and Model Selection

We select the best hyperparameters by searching a combination of batch size, learning rate and the number of fine-tuning epochs with the following range: learning rate $\{2 \times 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}\}$; batch size $\{16, 32\}$; number of epochs: $\{3, 4\}$. Note the best hyperparameters and models are selected by development performance in *English*.

	en	de	zh	es	fr	it	ja	ru	Average
In language supervised learning									
Schwenk and Li (2018)	92.2	93.7	87.3	94.5	92.1	85.6	85.4	85.7	89.5
mBERT	94.2	93.3	89.3	95.7	93.4	88.0	88.4	87.5	91.2
Zero-shot cross-lingual transfer									
Schwenk and Li (2018)	92.2	81.2	74.7	72.5	72.4	**69.4**	**67.6**	60.8	73.9
Artetxe and Schwenk (2019) ♠ †	89.9	**84.8**	71.9	**77.3**	**78.0**	**69.4**	60.3	67.8	**74.9**
mBERT	**94.2**	80.2	**76.9**	72.6	72.6	68.9	56.5	**73.7**	74.5

Table 3.2: MLDoc experiments. ♠ denotes the model is pretrained with bitext, and † denotes concurrent work. Bold and underline denote best and second best.

	en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	Average
Pseudo supervision with machine translated training data from English to target language																
Conneau and Lample (2019) (MLM+TLM) ♠ †	85.0	80.2	80.8	80.3	78.1	79.3	78.1	74.7	76.5	76.6	75.5	78.6	72.3	70.9	63.2	76.7
mBERT	82.1	76.9	78.5	74.8	72.1	75.4	74.3	70.6	70.8	67.8	63.2	76.2	65.3	65.3	60.6	71.6
Zero-shot cross-lingual transfer																
Conneau et al. (2018) (X-LSTM) ♠ ♢	73.7	67.7	68.7	67.7	68.9	67.9	65.4	64.2	64.8	66.4	64.1	65.8	64.1	55.7	58.4	65.6
Artetxe and Schwenk (2019) ♠ †	73.9	71.9	72.9	72.6	73.1	74.2	71.5	69.7	71.4	72.0	69.2	71.4	65.5	62.2	61.0	70.2
Conneau and Lample (2019) (MLM+TLM) ♠ ♢ †	85.0	78.7	78.9	77.8	76.6	77.4	75.3	72.5	73.1	76.1	73.2	76.5	69.6	68.4	67.3	75.1
Conneau and Lample (2019) (MLM) ♢ †	83.2	76.5	76.3	74.2	73.1	74.0	73.1	67.8	68.5	71.2	69.2	71.9	65.7	64.6	63.4	71.5
mBERT	82.1	73.8	74.3	71.1	66.4	68.9	69.0	61.6	64.9	69.5	55.8	69.3	60.0	50.4	58.0	66.3

Table 3.3: XNLI experiments. ♠ denotes the model is pretrained with cross-lingual signal including bitext or bilingual dictionary, † denotes concurrent work, and ♢ denotes model selection with target language dev set.

	en	nl	es	de	zh	Average (-en,-zh)
In language supervised learning						
Xie et al. (2018)	-	86.40	86.26	78.16	-	83.61
mBERT	91.97	90.94	87.38	82.82	93.17	87.05
Zero-shot cross-lingual transfer						
Xie et al. (2018) ♢	-	71.25	72.37	57.76	-	67.13
mBERT	91.97	**77.57**	**74.96**	**69.56**	51.90	**74.03**

Table 3.4: NER tagging experiments. ♢ denotes model selection with target language dev set.

lang	bg	da	de	en	es	fa	hu	it	nl	pl	pt	ro	sk	sl	sv	Average (-en)
In language supervised learning																
mBERT	99.0	97.9	95.2	97.1	97.1	97.8	96.9	98.7	92.1	98.5	98.3	97.8	97.0	98.9	98.4	97.4
Low resource cross-lingual transfer																
Kim et al. (2017) (1280)	95.7	94.3	90.7	-	93.4	94.8	94.5	95.9	85.8	92.1	95.5	94.2	90.0	94.1	94.6	93.3
Kim et al. (2017) (320)	92.4	90.8	89.7	-	90.9	91.8	90.7	94.0	82.2	85.5	94.2	91.4	83.2	90.6	90.7	89.9
Zero-shot cross-lingual transfer																
mBERT	87.4	88.3	89.8	97.1	85.2	72.8	83.2	84.7	75.9	86.9	82.1	84.7	83.6	84.2	91.3	84.3

Table 3.5: POS tagging. Kim et al. (2017) use small amounts of training data in the target language.

3.4 Is mBERT Multilingual?

3.4.1 MLDoc

We include two strong baselines. Schwenk and Li (2018) use MultiCCA, multilingual word embeddings trained with a bilingual dictionary (Ammar et al., 2016), and convolution neural networks. Concurrent to the publication of this chapter, Artetxe and Schwenk (2019) use bitext between English/Spanish and the rest of languages to pretrain a multilingual sentence representation with a sequence-to-sequence model where the decoder only has access to a max-pooling of the encoder hidden states.

mBERT outperforms (Table 3.2) multilingual word embeddings and performs comparably with a multilingual sentence representation, even though mBERT does not have access to bitext. Interestingly, mBERT outperforms Artetxe and Schwenk (2019) in distantly related languages like Chinese and Russian and under-performs in closely related Indo-European languages.

	Dist	mBERT(S)	Baseline(Z)	mBERT(Z)	mBERT(Z+POS)
en	0.00	91.5/81.3	90.4/**88.4**	91.5/81.3	**91.8**/82.2
no	0.06	93.6/85.9	80.8/**72.8**	80.6/68.9	**82.7**/72.1
sv	0.07	91.2/83.1	81.0/73.2	82.5/71.2	**84.3/73.7**
fr	0.09	91.7/85.4	77.9/72.8	82.7/72.7	**83.8/76.2**
pt	0.09	93.2/87.2	76.6/**67.8**	77.1/64.0	**78.3**/66.9
da	0.10	89.5/81.9	76.6/67.9	77.4/64.7	**79.3/68.1**
es	0.12	92.3/86.5	74.5/66.4	78.1/64.9	**79.0/68.9**
it	0.12	94.8/88.7	80.8/75.8	84.6/74.4	**86.0/77.8**
ca	0.13	94.3/89.5	73.8/65.1	78.1/64.6	**79.0/67.9**
hr	0.13	92.4/83.8	61.9/52.9	**80.7**/65.8	80.4/**68.2**
pl	0.13	94.7/79.9	74.6/62.2	82.8/59.4	**85.7/65.4**
sl	0.13	88.0/77.8	68.2/56.5	72.6/51.4	**75.9/59.2**
uk	0.13	90.6/83.4	60.1/52.3	**76.7**/60.0	76.5/**65.5**
bg	0.14	95.2/85.5	79.4/**68.2**	83.3/62.3	**84.4**/68.1
cs	0.14	94.2/86.6	63.1/53.8	76.6/58.7	**77.4/63.6**
de	0.14	86.1/76.5	71.3/61.6	80.4/66.3	**83.5/71.2**
he	0.14	91.9/83.6	55.3/48.0	**67.5**/48.4	67.0/**54.3**
nl	0.14	94.0/85.0	68.6/60.3	78.0/64.8	**79.9/67.1**
ru	0.14	94.7/88.0	60.6/51.6	**73.6**/58.5	73.2/**61.5**
ro	0.15	92.2/83.2	65.1/54.1	**77.0**/58.5	76.9/**62.6**
id	0.17	86.3/75.4	49.2/43.5	**62.6**/45.6	59.8/**48.6**
sk	0.17	93.8/83.3	66.7/58.2	82.7/63.9	**82.9/67.8**
lv	0.18	87.3/75.3	**70.8/49.3**	66.0/41.4	70.4/48.5
et	0.20	88.8/79.7	65.7/44.9	66.9/44.3	**70.8/50.7**
fi	0.20	91.3/81.8	66.3/48.7	68.4/47.5	**71.4/52.5**
zh*	0.23	88.3/81.2	42.5/25.1	**53.8**/26.8	53.4/**29.0**
ar	0.26	87.6/80.6	38.1/28.0	43.9/28.3	**44.7/32.9**
la	0.28	85.2/73.1	48.0/**35.2**	47.9/26.1	**50.9**/32.2
ko	0.33	86.0/74.8	34.5/16.4	**52.7**/27.5	52.3/**29.4**
hi	0.40	94.8/86.7	35.5/26.5	49.8/33.2	**58.9/44.0**
ja*	0.49	94.2/87.4	28.2/20.9	36.6/15.7	**41.3/30.9**
AVER	0.17	91.3/82.6	64.1/53.8	71.4/54.2	**73.0/58.9**

Table 3.6: Dependency parsing results by language (UAS/LAS). * denotes delexicalized parsing in the baseline. S and Z denotes supervised learning and zero-shot transfer. Bold and underline denotes best and second best. We order the languages by word order distance to English.

3.4.2 XNLI

We include three strong baselines, Artetxe and Schwenk (2019) and Conneau and Lample (2019) are concurrent to the publication of this chapter. Conneau and Lample (2019) with MLM is similar to mBERT; the main difference is that it only trains with the 15 languages of XNLI, has 249M parameters (around 40% more than mBERT), and MLM+TLM also uses bitext as training data [3]. Conneau et al. (2018) use supervised multilingual word embeddings with an LSTM encoder and max-pooling. After an English encoder and classifier are trained, the target encoder is trained to mimic the English encoder with ranking loss and bitext.

In Table 3.3, mBERT outperforms one model with bitext training but (as expected) falls short of models with more cross-lingual training information. Interestingly, mBERT and MLM are mostly the same except for the training languages, yet we observe that mBERT under-performs MLM by a large margin. We hypothesize that limiting pretraining to only those languages needed for the downstream task is beneficial. The gap between Artetxe and Schwenk (2019) and mBERT in XNLI is larger than MLDoc, likely because XNLI is harder.

3.4.3 NER

We use Xie et al. (2018) as a zero-shot cross-lingual transfer baseline, which is state-of-the-art on CoNLL 2002 and 2003. It uses unsupervised bilingual word embeddings (Lample et al., 2018) with a hybrid of a character-level/word-level LSTM, self-attention, and a CRF. Pseudo training data is built by word-to-word translation with an induced dictionary from

[3]They also use language embeddings as input and exclude the next sentence classification objective

bilingual word embeddings.

mBERT outperforms a strong baseline by an average of 6.9 points absolute F1 and an 11.8 point absolute improvement in German with a simple one layer 0^{th}-order CRF as a prediction function (Table 3.4). A large gap remains when transferring to distantly related languages (e.g. Chinese) compared to a supervised baseline. Further effort should focus on transferring between distantly related languages. In Section 3.7 we show that sharing subwords across languages helps transfer.

3.4.4 POS

We use Kim et al. (2017) as a reference. They utilized a small amount of supervision in the target language as well as English supervision so the results are not directly comparable. Table 3.5 shows a large (average) gap between mBERT and Kim et al. Interestingly, mBERT still outperforms Kim et al. (2017) with 320 sentences in German (de), Polish (pl), Slovak (sk) and Swedish (sv).

3.4.5 Dependency Parsing

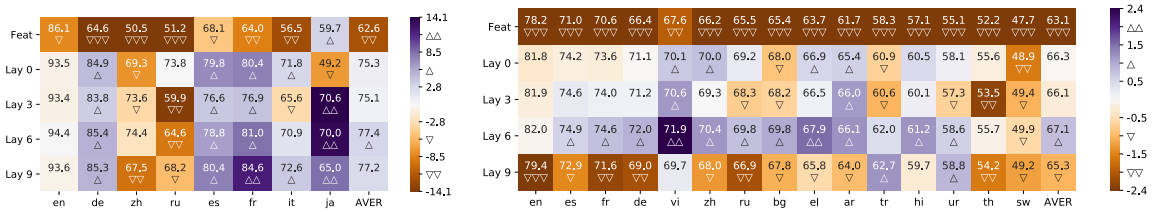
We use the best performing model on average in Ahmad et al. (2019) as a zero-shot transfer baseline, i.e. transformer encoder with graph-based parser (Dozat and Manning, 2017), and dictionary supervised cross-lingual embeddings (Smith et al., 2017). Dependency parsers, including Ahmad et al., assume access to gold POS tags: a cross-lingual representa-

tion. We consider two versions of mBERT: with and without gold POS tags. When tags are available, a tag embedding is concatenated with the final output of mBERT.

Table 3.6 shows that mBERT outperforms the baseline on average by 7.3 point UAS and 0.4 point LAS absolute improvement even without gold POS tags. Note in practice, gold POS tags are not always available, especially for low resource languages. Interestingly, the LAS of mBERT tends to be weaker than the baseline in languages with less word order distance, in other words, more closely related to English. With the help of gold POS tags, we further observe 1.6 points UAS and 4.7 point LAS absolute improvement on average. It appears that adding gold POS tags, which provide clearer cross-lingual representations, benefit mBERT.

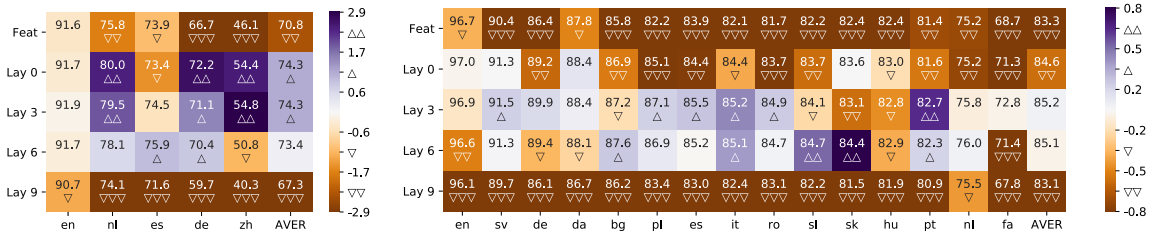
3.4.6 Summary

Across all five tasks, mBERT demonstrates strong (sometimes state-of-the-art) zero-shot cross-lingual performance without any cross-lingual signal. It outperforms cross-lingual embeddings in four tasks. With a small amount of target language supervision and cross-lingual signal, mBERT may improve further. In short, mBERT is a surprisingly effective cross-lingual model for many NLP tasks.



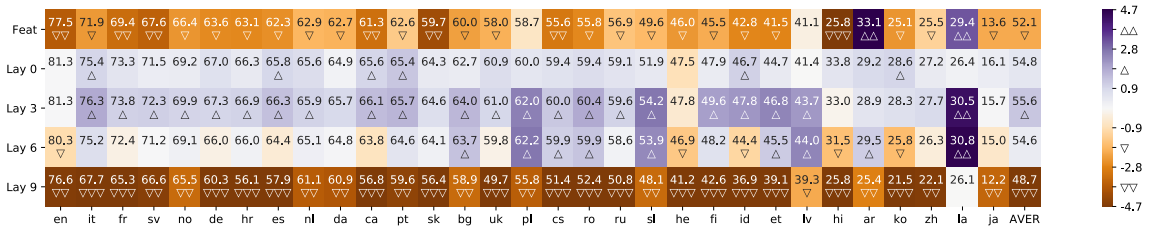
(a) Document classification (ACC)

(b) Natural language inference (ACC)



(c) NER (F1)

(d) POS tagging (ACC)



(e) Dependency parsing (LAS)

Figure 3.1: Performance of different fine-tuning approaches compared with fine-tuning all mBERT parameters. Color denotes absolute difference and the number in each entry is the evaluation in the corresponding setting. Languages are sorted by mBERT zero-shot transfer performance. Three downward triangles indicate performance drop more than the legend's lower limit.

3.5 Does mBERT Vary Layer-wise?

The goal of a deep neural network is to abstract to higher-order representations as you progress up the hierarchy (Yosinski et al., 2014). Peters et al. (2018) empirically show that for ELMo in English the lower layer is better at syntax while the upper layer is better at semantics. However, it is unclear how different layers affect the quality of cross-lingual

representation. For mBERT, we hypothesize a similar generalization across the 13 layers, as well as an abstraction away from a specific language with higher layers. Does the zero-shot transfer performance vary with different layers?

We consider two schemes. First, we follow the feature-based approach of ELMo by taking a learned weighted combination of all 13 layers of mBERT with a two-layer bidirectional LSTM with d_h hidden size (Feat). Note the LSTM is trained from scratch and mBERT is fixed. For sentence and document classification, an additional max-pooling is used to extract a fixed-dimension vector. We train the feature-based approach with Adam and learning rate 1e-3. The batch size is 32. The learning rate is halved whenever the development evaluation does not improve. The training is stopped early when learning rate drops below 1e-5. Second, when fine-tuning mBERT, we fix the bottom n layers (n included) of mBERT, where layer 0 is the input embedding. We consider $n \in \{0, 3, 6, 9\}$.

Freezing the bottom layers of mBERT, in general, improves the performance of mBERT in all five tasks (Figure 3.1). For sentence-level tasks like document classification and natural language inference, we observe the largest improvement with $n = 6$. For word-level tasks like NER, POS tagging, and parsing, we observe the largest improvement with $n = 3$. More improvement in under-performing languages is observed.

In each task, the feature-based approach with LSTM under-performs the fine-tuning approach. We hypothesize that initialization from pretraining with lots of languages provides a very good starting point that is hard to beat. Additionally, the LSTM could also be part of the problem. In Ahmad et al. (2019) for dependency parsing, an LSTM encoder was

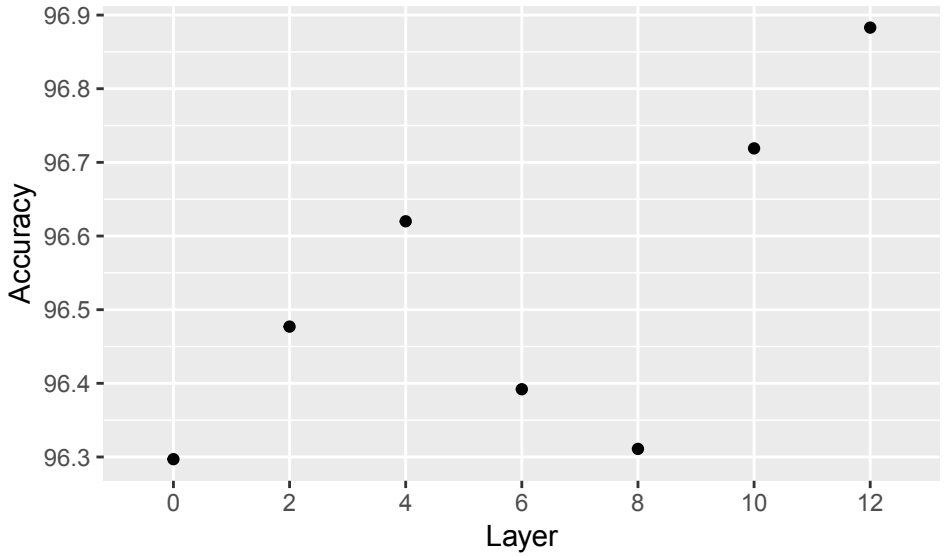


Figure 3.2: Language identification accuracy for different layer of mBERT. layer 0 is the embedding layer and the layer $i > 0$ is the output of the i$^{\text{th}}$ transformer block.

worse than a transformer when transferring to languages with high word ordering distance to English.

3.6 Does mBERT Retain Language Specific Information?

mBERT may learn a cross-lingual representation by abstracting away from language-specific information, thus losing the ability to distinguish between languages. We test this by considering language identification: does mBERT retain language-specific information? We use WiLI-2018 (Thoma, 2018), which includes over 200 languages from Wikipedia. We

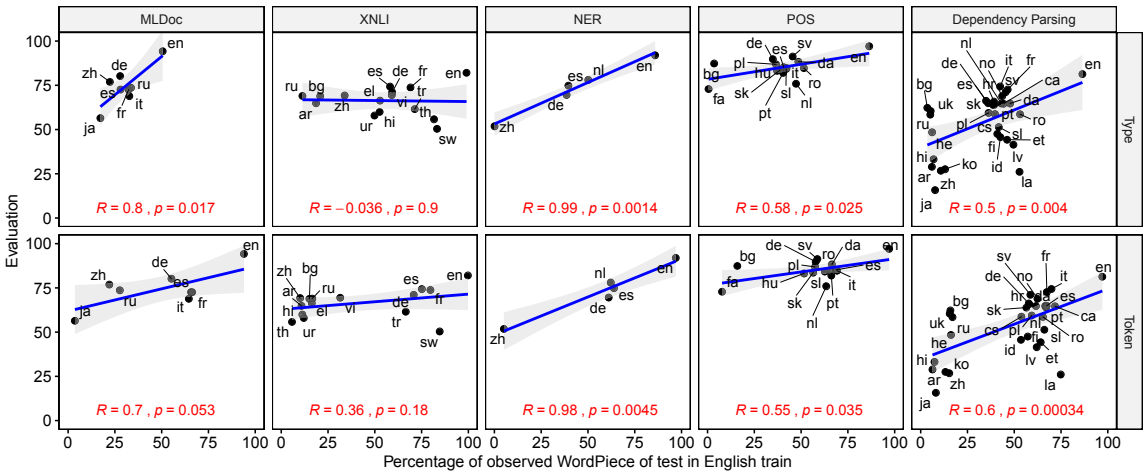


Figure 3.3: Relation between cross-lingual zero-shot transfer performance with mBERT and percentage of observed subwords at both type-level and token-level. Pearson correlation coefficient and p-value are shown in red.

keep only those languages included in mBERT, leaving 99 languages [4]. We take various layers of bag-of-words mBERT representation of the first two sentences of the test paragraph and add a linear classifier with softmax. We fix mBERT and train *only* the classifier the same as the feature-based approach in Section 3.5.

All tested layers achieved around 96% accuracy (Figure 3.2), with no clear difference between layers. This suggests each layer contains language-specific information; surprising given the zero-shot cross-lingual abilities. As mBERT generalizes its representations and creates cross-lingual representations, it maintains language-specific details. This may be encouraged during pretraining since mBERT needs to retain enough language-specific information to perform the cloze task.

[4]Hungarian, Western-Punjabi, Norwegian-Bokmal, and Piedmontese are not covered by WiLI.

3.7 Does mBERT Benefit by Sharing Subwords Across Languages?

As discussed in Section 2.8, mBERT shares subwords in closely related languages or perhaps in distantly related languages. At training time, the representation of a shared subword is explicitly trained to contain enough information for the cloze task in all languages in which it appears. During fine-tuning for zero-shot cross-lingual transfer, if a subword in the target language test set also appears in the source language training data, the supervision could be leaked to the target language explicitly. However, all subwords interact in a non-interpretable way inside a deep network, it is hard to characterize how sharing subwords affects the transfer performance. Additionally, subword representations could overfit to the source language and potentially hurt transfer performance. In these experiments, we investigate how sharing subwords across languages affects cross-lingual transfer.

To quantify how many subwords are shared across languages in any task, we assume $V^{\text{en}}_{\text{train}}$ is the set of all subwords in the English training set, V^{ℓ}_{test} is the set of all subwords in language ℓ test set, and c^{ℓ}_{w} is the count of subword w in test set of language ℓ. We then calculate the percentage of observed subwords at type-level p^{ℓ}_{type} and token-level p^{ℓ}_{token} for

each target language ℓ.

$$p^{\ell}_{\text{type}} = \frac{|V^{\ell}_{\text{obs}}|}{|V^{\ell}_{\text{test}}|} \times 100 \tag{3.1}$$

$$p^{\ell}_{\text{token}} = \frac{\sum_{w \in V^{\ell}_{\text{obs}}} c^{\ell}_{w}}{\sum_{w \in V^{\ell}_{\text{test}}} c^{\ell}_{w}} \times 100 \tag{3.2}$$

where $V^{\ell}_{\text{obs}} = V^{\text{en}}_{\text{train}} \cap V^{\ell}_{\text{test}}$.

In Figure 3.3, we show the relation between cross-lingual zero-shot transfer performance of mBERT and p^{ℓ}_{type} or p^{ℓ}_{token} for all five tasks with Pearson correlation. In four out of five tasks (not XNLI) we observed a strong positive correlation ($p < 0.05$) with a correlation coefficient larger than 0.5. In Indo-European languages, we observed p^{ℓ}_{token} is usually around 50% to 75% while p^{ℓ}_{type} is usually less than 50%. This indicates that subwords shared across languages are usually high frequency[5].

3.8 Discussion

In this chapter, we show mBERT does well in a cross-lingual zero-shot transfer setting on five different tasks covering a large number of languages, even without any explicit cross-lingual signal during pretraining. It outperforms cross-lingual embeddings, which typically have more cross-lingual supervision. By fixing the bottom layers of mBERT during fine-tuning, we observe further performance gains. Language-specific information is

[5]With the data-dependent WordPiece algorithm, subwords that appear in multiple languages with high frequency are more likely to be selected.

preserved in all layers. Sharing subwords helps cross-lingual transfer; a strong correlation is observed between the percentage of overlapping subwords and transfer performance. mBERT effectively learns a good multilingual representation with strong cross-lingual zero-shot transfer performance in various tasks. Even without explicit cross-lingual supervision, these models do very well.

This thesis builds on top of these findings. In Section 3.7, we observe a correlation between subword overlap between languages and cross-lingual transfer performance. However, this is surprisingly not causation despite being intuitive, as we will show in chapter 4, determining which factor contributes the most to the learning of cross-lingual representation. While we experimented with 39 languages in this chapter, the majority of languages supported by mBERT are still untested. In chapter 5, we test the low resource languages within mBERT. As we show with XNLI in Section 3.4, while bitext is hard to obtain in low resource settings, a variant of mBERT pretrained with bitext (Conneau and Lample, 2019) shows even stronger performance. In chapter 6, we explore how to introduce cross-lingual supervision into models like BERT. With POS tagging in Section 3.4, we show mBERT, in general, under-performs models with a small amount of supervision. Lauscher et al. (2020) shows few-shot cross-lingual transfer improves zero-shot cross-lingual transfer, although the choice of shot has significant impact on the performance (Zhao et al., 2021). Such observation is not surprising, as we will take a deeper dive in chapter 7 looking at why zero-shot cross-lingual transfer has high variance. In chapter 8, we explore how to construct better data projection pipeline to improve zero-shot cross-lingual transfer with multilingual

models like mBERT.

Outside of this thesis, many papers build on top of these findings. By scaling up mBERT with bigger dataset and model, better cross-lingual representation can be achieved, including models like XLM-R (Conneau et al., 2020a), mT5 (Xue et al., 2021), and XLM-R_{XXL} (Goyal et al., 2021). With strong cross-lingual representation covering over 100 languages, mBERT enables massively multilingual models like multilingual parser UDify (Kondratyuk and Straka, 2019). As more and more multilingual models become available, benchmarks have been introduced aggregating existing multilingual dataset (Hu et al., 2020; Liang et al., 2020).

Chapter 4

How Does mBERT Learn Cross-lingual Representation?

4.1 Introduction

In chapter 3, we observe that multilingual language models such as mBERT (Devlin et al., 2019) and XLM (Conneau and Lample, 2019) surprisingly enable effective cross-lingual transfer—it is possible to learn a model from supervised data in one language and apply it to another with no additional training—for a wide range of tasks without any explicit cross-lingual signal. chapter 3 observes that there is a positive correlation between vocabulary overlap between languages and transfer performance across languages. However, without controlled experiment, it is unclear whether such correlation is causal. More broadly, it is unclear why models like mBERT learn cross-lingual representation without any explicit cross-lingual signal.

In this chapter, we first present a detailed ablation study on the impact of each modeling decision of pretraining on the learning of cross-lingual representation. We look at factors including domain similarity of pretraining corpus, a single shared subword vocabulary across languages, vocabulary overlap between languages, random word replacement during pretraining, joint softmax prediction across languages, and transformer parameter sharing across languages.

Much to our surprise, we discover that pretrained models still learn cross-lingual representation without any shared vocabulary or domain similarity, and even when only a subset of the parameters in the joint encoder are shared. In particular, by systematically varying the amount of shared vocabulary between two languages during pretraining, we show that the amount of overlap only accounts for a few points of performance in transfer tasks, much

less than might be expected. By sharing transformer parameters alone, pretraining learns to map similar words and sentences to similar hidden representations.

How does sharing transformer parameters alone allow a model to learn cross-lingual representation? To better understand these observation, we also analyze multiple monolingual BERT models trained independently. We find that monolingual models trained in different languages learn representations that align with each other surprisingly well, even though they have no shared parameters during pretraining and completely different vocabulary. This result closely mirrors the widely observed fact that word embeddings can be effectively aligned across languages (Mikolov, Le, and Sutskever, 2013). Similar dynamics are at play in multilingual pretraining. As monolingual BERTs of different language are similar to each other, when the transformer parameters are shared across languages during pretraining, the multilingual model naturally align the representation of different language in a cross-lingual fashion.

4.2 Background

4.2.1 Alignment of Embeddings

In Section 2.7, we discuss the alignment of monolingual word embeddings and ELMo to produce cross-lingual representation. Wang et al. (2019) align mBERT representations and evaluate on dependency parsing.

4.2.2 Neural Network Activation Similarity

We hypothesize that similar to word embedding spaces, language-universal structures emerge in pretrained language models. While computing word embedding similarity is relatively straightforward, the same cannot be said for the deep contextualized BERT models that we study. Researchers introduce ways to measure the similarity of neural network activation between different layers and different models (Laakso and Cottrell, 2000; Li et al., 2016; Raghu et al., 2017; Morcos, Raghu, and Bengio, 2018; Wang et al., 2018). For example, Raghu et al. (2017) use canonical correlation analysis (CCA) and a new method, singular vector canonical correlation analysis (SVCCA), to show that early layers converge faster than upper layers in convolutional neural networks. Kudugunta et al. (2019) use SVCCA to investigate the multilingual representations obtained by the encoder of a massively multilingual neural machine translation system (Aharoni, Johnson, and Firat, 2019). Kornblith et al. (2019) argue that CCA fails to measure meaningful similarities between representations that have a higher dimension than the number of data points and introduce the centered kernel alignment (CKA) to solve this problem. They successfully use CKA to identify correspondences between activations in networks trained from different initializations.

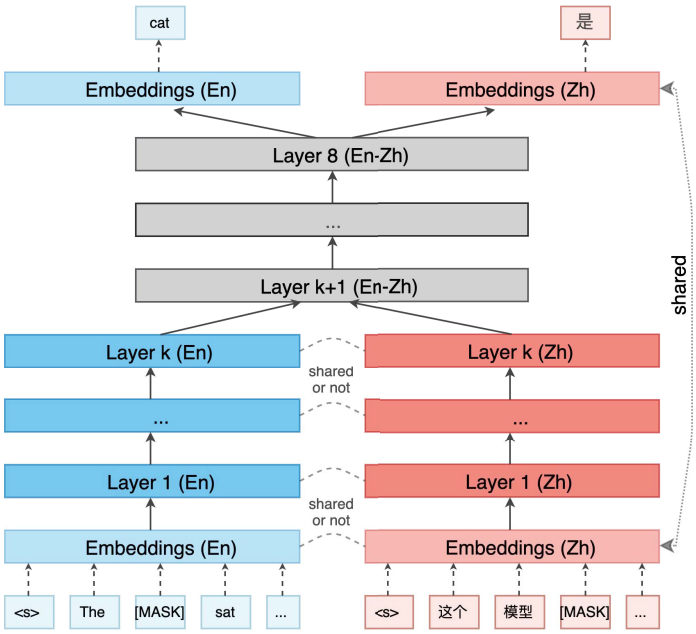


Figure 4.1: On the impact of anchor points and parameter sharing on the emergence of multilingual representations. We train bilingual masked language models and remove parameter sharing for the embedding layers and first few Transformers layers to probe the impact of anchor points and shared structure on cross-lingual transfer.

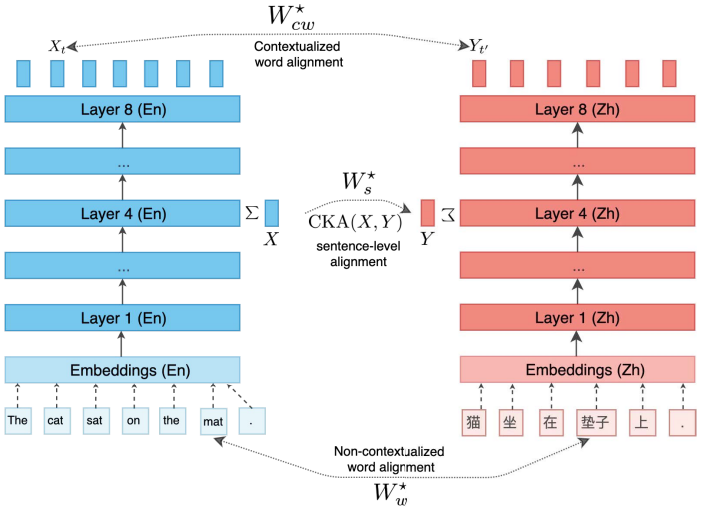


Figure 4.2: Probing the layer similarity of monolingual BERT models. We investigate the similarity of separate monolingual BERT models at different levels. We use an orthogonal mapping between the pooled representations of each model. We also quantify the similarity using the centered kernel alignment (CKA) similarity index.

4.3 Cross-lingual Pretraining

We study a standard multilingual masked language modeling formulation and evaluate performance on several different cross-lingual transfer tasks, as described in this section.

4.3.1 Multilingual Masked Language Modeling

Our multilingual masked language models follow the setup used by both mBERT and XLM. We use the implementation of Conneau and Lample (2019). Specifically, we consider continuous streams of 256 tokens and mask 15% of the input tokens which we replace 80% of the time by a mask token, 10% of the time with the original word, and 10% of the time with a random word. Note the random words could be foreign words. The model is trained to recover the masked tokens from its context (Taylor, 1953). The subword vocabulary and model parameters are shared across languages. Note the model has a softmax prediction layer shared across languages. We use Wikipedia for training data, preprocessed by Moses (Koehn et al., 2007) and Stanford word segmenter (for Chinese only) and BPE (Sennrich, Haddow, and Birch, 2016) to learn subword vocabulary. During training, we sample a batch of continuous streams of text from one language proportionally to the fraction of sentences in each training corpus, exponentiated to the power 0.7.

4.3.2 Pretraining Details

Each model is a Transformer (Vaswani et al., 2017) with 8 layers, 12 heads and GELU activiation functions (Hendrycks and Gimpel, 2016). The output softmax layer is tied with input embeddings (Press and Wolf, 2017). The embeddings dimension is 768, the hidden dimension of the feed-forward layer is 3072, and dropout is 0.1. We train our models with the Adam optimizer (Kingma and Ba, 2014) and the inverse square root learning rate scheduler of Vaswani et al. (2017) with 10^{-4} learning rate and 30k linear warm up steps. For each model, we train it with 8 NVIDIA V100 GPUs with 32GB of memory and mixed precision. It takes around 3 days to train one model. We use batch size 96 for each GPU and each epoch contains 200k batches. We stop training at epoch 200 and select the best model based on English dev perplexity for evaluation.

4.4 Cross-lingual Evaluation

We consider three NLP tasks to evaluate performance: natural language inference (NLI), named entity recognition (NER) and dependency parsing (Parsing). Similar to chapter 3, we adopt the **zero-shot cross-lingual transfer** setting, where we (1) fine-tune the pretrained model on English and (2) directly transfer the model to target languages. We select the model and tune hyperparameters with the English dev set. We report the result on average of the best two sets of hyperparameters.

4.4.1 Fine-tuning Details

We fine-tune the model for 10 epochs for NER and Parsing and 200 epochs for NLI. We search the following hyperparameter for NER and Parsing: batch size $\{16, 32\}$; learning rate $\{$2e-5, 3e-5, 5e-5$\}$. For XNLI, we search: batch size $\{4, 8\}$; encoder learning rate $\{$1.25e-6, 2.5e-6, 5e-6$\}$; classifier learning rate $\{$5e-6, 2.5e-5, 1.25e-4$\}$. We use Adam with a fixed learning rate for XNLI and warmup the learning rate for the first 10% batch then decrease linearly to 0 for NER and Parsing. We save a checkpoint after each epoch.

4.4.2 Natural Language Inference

We use the cross-lingual natural language inference (XNLI) dataset (Conneau et al., 2018). The task-specific layer is a linear mapping to a softmax classifier, which takes the representation of the first token as input.

4.4.3 Named Entity Recognition

We use WikiAnn (Pan et al., 2017), a silver NER dataset built automatically from Wikipedia, for English-Russian and English-French. For English-Chinese, we use CoNLL 2003 English (Tjong Kim Sang and De Meulder, 2003) and a Chinese NER dataset (Levow, 2006), with realigned Chinese NER labels based on the Stanford word segmenter. We model NER as BIO tagging. Similar to Section 3.2.3, the task-specific layer is a linear mapping to a softmax classifier, which takes the representation of the first subword of each word

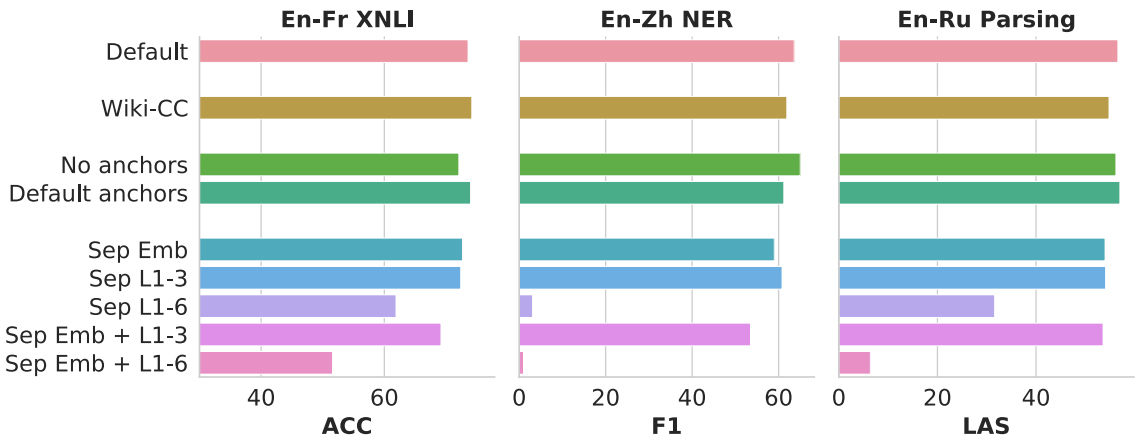


Figure 4.3: Cross-lingual transfer of bilingual MLM on three tasks and language pairs under different settings. Other tasks and language pairs follow similar trends. See Table 4.1 for full results.

as input. We report span-level F1. We adopt the same post-processing heuristic steps as Section 3.2.3.

4.4.4 Dependency Parsing

Finally, we use the Universal Dependencies (UD v2.3) (Nivre, 2018b) for dependency parsing. We consider the following four treebanks: English-EWT, French-GSD, Russian-GSD, and Chinese-GSD. The task-specific layer is a graph-based parser (Dozat and Manning, 2017), using representations of the first subword of each word as inputs, same as Section 3.2.5. We measure performance with the labeled attachment score (LAS).

Model	Domain	BPE Merges	Anchors Pts	Share Param.	Softmax	XNLI (Acc)				NER (F1)				Parsing (LAS)			
						fr	ru	zh	Δ	fr	ru	zh	Δ	fr	ru	zh	Δ
Default	Wiki-Wiki	80k	all	all	shared	73.6	68.7	68.3	0.0	79.8	60.9	63.6	0.0	73.2	56.6	28.8	0.0
Domain Similarity (Section 4.5.1)																	
Wiki-CC	Wiki-CC	-	-	-	-	74.2	65.8	66.5	-1.4	74.0	49.6	61.9	-6.2	71.3	54.8	25.2	-2.5
Anchor Points (Section 4.5.2)																	
No anchors	-	40k/40k	0	-	-	72.1	67.5	67.7	-1.1	74.0	57.9	65.0	-2.4	72.3	56.2	27.4	-0.9
Default anchors	-	40k/40k	-	-	-	74.0	68.1	68.9	+0.1	76.8	56.3	61.2	-3.3	73.0	57.0	28.3	-0.1
Parameter Sharing (Section 4.5.3)																	
Sep Emb	-	40k/40k	0*	Sep Emb	lang-specific	72.7	63.6	60.8	-4.5	75.5	57.5	59.0	-4.1	71.7	54.0	27.5	-1.8
Sep L1-3	-	40k/40k	-	Sep L1-3	-	72.4	65.0	63.1	-3.4	74.0	53.3	60.8	-5.3	69.7	54.1	26.4	-2.8
Sep L1-6	-	40k/40k	-	Sep L1-6	-	61.9	43.6	37.4	-22.6	61.2	23.7	3.1	-38.7	61.7	31.6	12.0	-17.8
Sep Emb + L1-3	-	40k/40k	0*	Sep Emb + L1-3	lang-specific	69.2	61.7	56.4	-7.8	73.8	46.8	53.5	-10.0	68.2	53.6	23.9	-4.3
Sep Emb + L1-6	-	40k/40k	0*	Sep Emb + L1-6	lang-specific	51.6	35.8	34.4	-29.6	56.5	5.4	1.0	-47.1	50.9	6.4	1.5	-33.3

Table 4.1: Dissecting bilingual MLM based on zero-shot cross-lingual transfer performance. - denote the same as the first row (**Default**). Δ denote the difference of average task performance between a model and **Default**.

4.5 What Makes mBERT Multilingual?

We hypothesize that the following factors play important roles in what makes multilingual BERT multilingual: domain similarity, shared vocabulary (or anchor points), shared parameters, and language similarity. Without loss of generality, we focus on bilingual MLM. We consider three pairs of languages with different levels of language similarity: English-French, English-Russian, and English-Chinese.

4.5.1 Domain Similarity

Multilingual BERT and XLM are trained on the Wikipedia comparable corpora. Domain similarity has been shown to affect the quality of cross-lingual word embeddings (Lample et al., 2018), but this effect is not well established for masked language models. We consider domain differences by training on Wikipedia for English and a random subset of Common Crawl of the same size for the other languages (**Wiki-CC**). We also consider a model trained

with Wikipedia only (**Default**) for comparison.

The first group in Table 4.1 shows domain mismatch has a relatively modest effect on performance. XNLI and parsing performance drop around 2 points while NER drops over 6 points for all languages on average. One possible reason is that the labeled WikiAnn data for NER consists of Wikipedia text; domain differences between source and target language during pretraining hurt performance more. Indeed for English and Chinese NER, where neither side comes from Wikipedia, performance only drops around 2 points.

4.5.2 Anchor Points

Anchor points are *identical strings* that appear in both languages in the training corpus. Translingual words like *DNA* or *Paris* appear in the Wikipedia of many languages with the same meaning. In mBERT, anchor points are naturally preserved due to joint BPE and shared vocabulary across languages. Anchor point existence has been suggested as a key ingredient for effective cross-lingual transfer since they allow the shared encoder to have at least some direct tying of meaning across different languages (Conneau and Lample, 2019; Pires, Schlinger, and Garrette, 2019; Wu and Dredze, 2019). However, this effect has not been carefully measured.

We present a controlled study of the impact of anchor points on cross-lingual transfer performance by varying the amount of shared subword vocabulary across languages. Instead of using a single joint BPE with 80k merges, we use language-specific BPE with 40k merges for each language. We then build vocabulary by taking the union of the vocabulary of two

languages and train a bilingual MLM (**Default anchors**). To remove anchor points, we add a language prefix to each word in the vocabulary before taking the union. Bilingual MLM (**No anchors**) trained with such data has no shared vocabulary across languages. However, it still has a single softmax prediction layer shared across languages and tied with input embeddings.

The second group of Table 4.1 shows cross-lingual transfer performance under the two anchor point conditions. Surprisingly, effective transfer is still possible with no anchor points. Comparing no anchors and default anchors, the performance of XNLI and parsing drops only around 1 point while NER even improves 1 point averaging over three languages. Overall, these results show that we have previously overestimated the contribution of anchor points during multilingual pretraining. Concurrent to the publication of this chapter, K et al. (2020) similarly find anchor points play a minor role in learning cross-lingual representation.

4.5.3 Parameter Sharing

Given that anchor points are not required for transfer, a natural next question is the extent to which we need to tie the parameters of the transformer layers. Sharing the parameters of the top layer is necessary to provide shared inputs to the task-specific layer. However, as seen in Figure 4.1, we can progressively separate the *bottom* layers 1:3 and 1:6 of the Transformers and/or the embedding layers (including positional embeddings) (**Sep Emb**; **Sep L1-3**; **Sep L1-6**; **Sep Emb + L1-3**; **Sep Emb + L1-6**). Since the prediction layer is tied with the embeddings layer, separating the embeddings layer also introduces a language-specific

softmax prediction layer for the cloze task. This effectively introduces language specific component into the multilingual model. In theory, such language specific component might learn to encode language specific information into a shared space, benefiting the learning of cross-lingual representation. Additionally, in this group of experiment, we only sample random words within one language during the MLM pretraining, as MLM pretraining would potentially introduce accidental anchor points during random word replacement. During fine-tuning on the English training set, we freeze the language-specific layers and only fine-tune the shared layers.

The third group in Table 4.1 shows cross-lingual transfer performance under different parameter sharing conditions with "Sep" denoting which layers **is not** shared across languages. Sep Emb (effectively no anchor point) drops more than No anchors with 3 points on XNLI and around 1 point on NER and parsing, suggesting having a cross-language softmax layer also helps to learn cross-lingual representations. Performance degrades as fewer layers are shared for all pairs, and again the less closely related language pairs lose the most. Most notably, the cross-lingual transfer performance drops to random when separating embeddings and bottom 6 layers of the transformer. However, reasonably strong levels of transfer are still possible without tying the bottom three layers. These trends suggest that parameter sharing is the key ingredient that enables the learning of an effective cross-lingual representation space, and having language-specific capacity does not help learn a language-specific encoder for cross-lingual representation despite having extra parameters.

4.5.4 Language Similarity

Finally, in contrast to many of the experiments above, language similarity seems to be quite important for effective transfer. Looking at Table 4.1 column by column in each task, we observe performance drops as language pairs become more distantly related. The more complex tasks seem to have larger performance gaps and having language-specific capacity does not seem to be the solution.

4.5.5 Conclusion

Summarised by Figure 4.3, parameter sharing is the most important factor. Anchor points and shared softmax projection parameters are not necessary for effective cross-lingual transfer. Joint BPE and domain similarity contribute a little in learning cross-lingual representation.

4.6 How Does Parameter Sharing Enable Cross-lingual Representation?

In Section 4.5, we observe that parameter sharing is the key for learning cross-lingual representation. How does parameter sharing enable cross-lingual representation? Our hypothesis is that the representations that the models learn for different languages are similarly shaped and during multilingual pretraining, the models naturally align its representation

across languages. If the hypothesis were true, we would be able to show that independently trained monolingual BERT models learn representations that are similar across languages, much like the widely observed similarities in word embedding spaces.

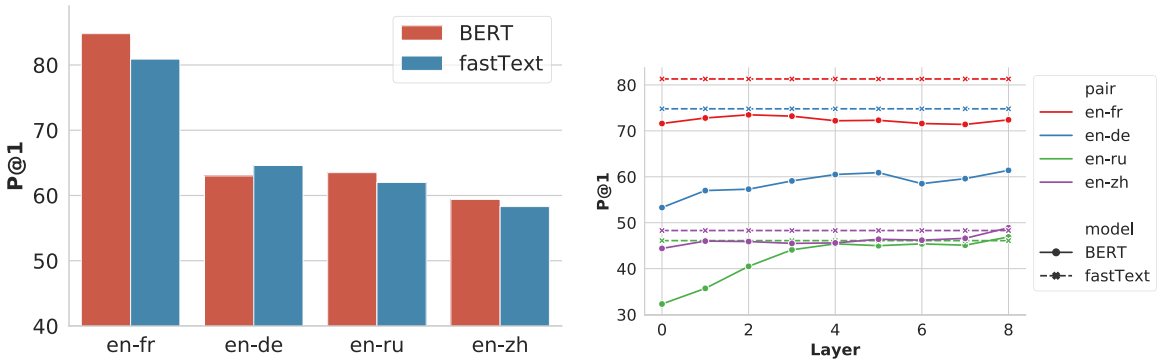
In this section, we show that independent monolingual BERT models produce highly similar representations when evaluated at the word level (Section 4.6.1.1), contextual word-level (Section 4.6.1.2), and sentence level (Section 4.6.1.3) . We also plot the cross-lingual similarity of neural network activation with center kernel alignment (Section 4.6.2) at each layer. We consider five languages: English, French, German, Russian, and Chinese.

4.6.1 Aligning Monolingual BERTs

To measure similarity, we learn an orthogonal mapping using the Procrustes (Smith et al., 2017) approach:

$$W^\star = \underset{W \in O_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_F = UV^T \tag{4.1}$$

with $U\Sigma V^T = \text{SVD}(YX^T)$, where X and Y are representations of two monolingual BERT models, sampled at different granularities as described below. We apply iterative normalization on X and Y before learning the mapping (Zhang et al., 2019).



(a) Non-contextual word embeddings alignment

(b) Contextual word embedding alignment

Figure 4.4: Alignment of word-level representations from monolingual BERT models on a subset of MUSE benchmark. Figure 4.4a and Figure 4.4b are not comparable due to different embedding vocabularies.

4.6.1.1 Word-level Alignment

In this section, we align both the non-contextual word representations from the embedding layers, and the contextual word representations from the hidden states of the Transformer at each layer.

For non-contextualized word embeddings, we define X and Y as the word embedding layers of monolingual BERT, which contain a single embedding per word (type). Note that in this case we only keep words containing only one subword. For contextualized word representations, we first encode 500k sentences in each language. At each layer, and for each word, we collect all contextualized representations of a word in the 500k sentences and average them to get a single embedding. Since BERT operates at the subword level, for one word we consider the average of all its subword embeddings. Eventually, we get one word embedding per layer. We use the MUSE benchmark (Lample et al., 2018), a bilingual dictionary induction dataset for alignment supervision and evaluate the alignment

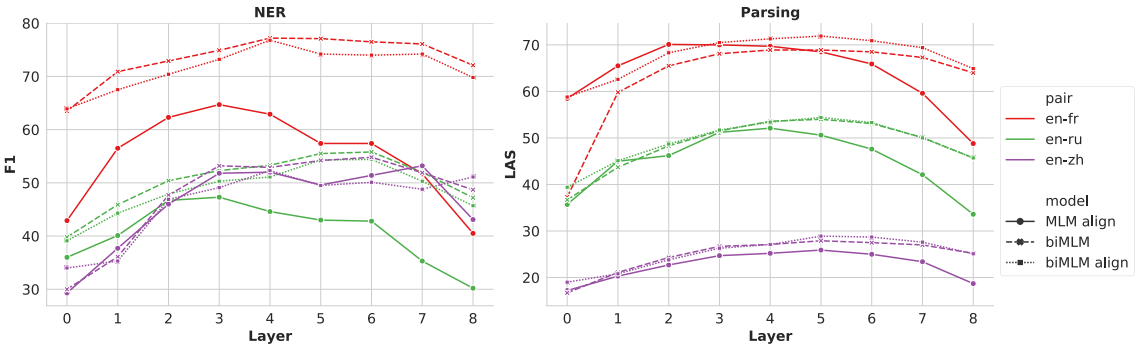


Figure 4.5: Contextual representation alignment of different layers for zero-shot cross-lingual transfer.

on word translation retrieval. As a baseline, we use the first 200k embeddings of fastText (Bojanowski et al., 2017) and learn the mapping using the same procedure as Section 4.6.1. Note we use a subset of 200k vocabulary of fastText, the same as BERT, to get a comparable number. We retrieve word translation using CSLS (Lample et al., 2018) with K=10.

In Figure 4.4, we report the alignment results under these two settings. Figure 4.4a shows that the subword embeddings matrix of BERT, where each subword is a standalone word, can easily be aligned with an orthogonal mapping and obtain slightly better performance than the same subset of fastText. Figure 4.4b shows embeddings matrix with the average of all contextual embeddings of each word can also be aligned to obtain a decent quality bilingual dictionary, although underperforming fastText. We notice that using contextual representations from higher layers obtain better results compared to lower layers.

4.6.1.2 Contextual Word-level Alignment

In addition to aligning word representations, we also align representations of two monolingual BERT models in contextual settings, and evaluate performance on cross-lingual transfer for NER and parsing. We take the Transformer layers of each monolingual model up to layer i, and learn a mapping W from layer i of the target model to layer i of the source model. To create that mapping, we use the same Procrustes approach but use a dictionary of parallel contextual words, obtained by running the fastAlign (Dyer, Chahuneau, and Smith, 2013) model on the 10k XNLI parallel sentences.

For each downstream task, we learn task-specific layers on top of i-th English layer: four Transformer layers and a task-specific layer. We learn these on the training set, but keep the first i pretrained layers freezed. After training these task-specific parameters, we encode (say) a Chinese sentence with the first i layers of the target Chinese BERT model, project the contextualized representations back to the English space using the W we learned, and then use the task-specific layers for NER and parsing.

In Figure 4.5, we vary i from the embedding layer (layer 0) to the last layer (layer 8) and present the results of our approach on parsing and NER. We also report results using the first i layers of a bilingual MLM (biMLM), and the same alignment step with biMLM. We show that aligning monolingual models (MLM align) obtain relatively good performance even though they perform worse than bilingual MLM, except for parsing in English-French. However, the same alignment step with biMLM only shows improvement in parsing. The results of monolingual alignment generally show that we can align contextual

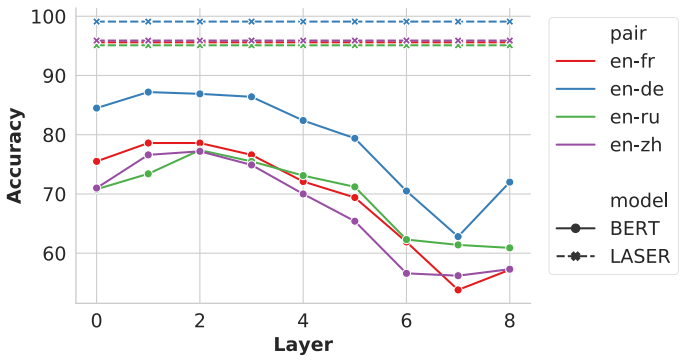


Figure 4.6: Parallel sentence retrieval accuracy after Procrustes alignment of monolingual BERT models.

representations of monolingual BERT models with a simple linear mapping and use this approach for cross-lingual transfer. We also observe that the model obtains the highest transfer performance with the middle layer representation alignment, and not the last layers. The performance gap between monolingual MLM alignment and bilingual MLM is higher in NER compared to parsing, suggesting the syntactic information needed for parsing might be easier to align with a simple mapping while entity information requires more explicit entity alignment.

4.6.1.3 Sentence-level Alignment

In this case, X and Y are obtained by average pooling subword representation (excluding special token) of sentences *at each layer* of monolingual BERT. We use multi-way parallel sentences from XNLI for alignment supervision and Tatoeba (Artetxe and Schwenk, 2019) for evaluation.

Figure 4.6 shows the sentence similarity search results with nearest neighbor search

and cosine similarity, evaluated by precision at 1, with four language pairs. Here the best result is obtained at lower layers. The performance is surprisingly good given we only use 10k parallel sentences to learn the alignment without fine-tuning at all. As a reference, the state-of-the-art performance is over 95%, obtained by LASER (Artetxe and Schwenk, 2019) trained with millions of parallel sentences.

4.6.1.4 Conclusion

These findings demonstrate that both word-level, contextual word-level, and sentence-level BERT representations can be aligned with a simple orthogonal mapping. Similar to the alignment of word embeddings (Mikolov, Le, and Sutskever, 2013), this shows that BERT models are similar across languages. This result gives more intuition on why mere parameter sharing is sufficient for multilingual representations to emerge in multilingual masked language models.

4.6.2 Neural Network Similarity

Based on the work of Kornblith et al. (2019), we examine the centered kernel alignment (CKA), a neural network similarity index that improves upon canonical correlation analysis (CCA), and use it to measure the similarity across both monolingual and bilingual masked language models. The linear CKA is both invariant to orthogonal transformation and isotropic scaling, but are not invertible to any linear transform. The linear CKA similarity

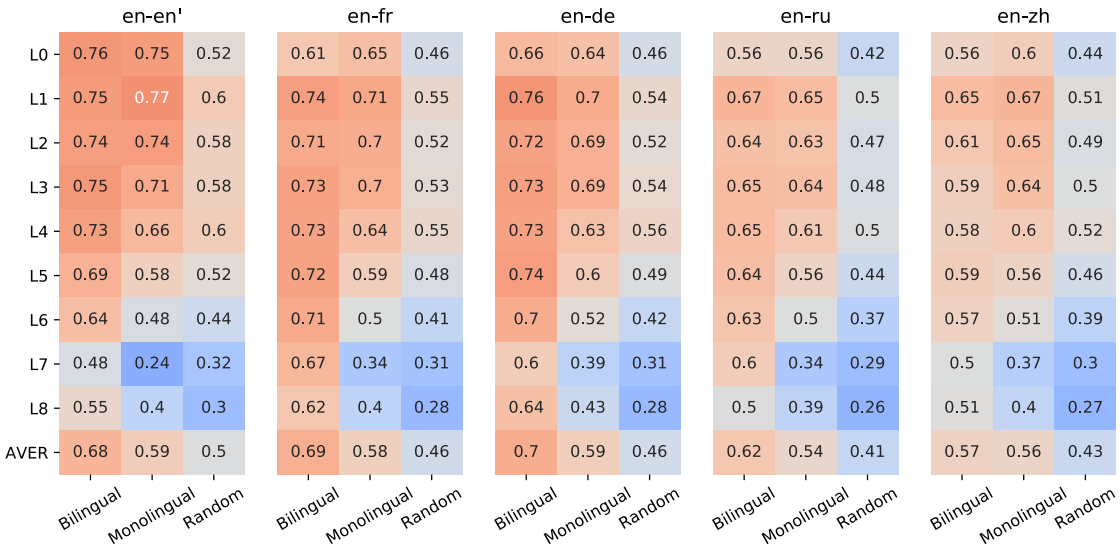


Figure 4.7: CKA similarity of mean-pooled multi-way parallel sentence representation at each layer. Note en′ corresponds to paraphrases of en obtained from back-translation (en-fr-en′). Random encoder is only used by non-Engligh sentences. L0 is the embedding layer while L1 to L8 are the corresponding transformer layers. The average row is the average of 9 (L0-L8) similarity measurements.

measure is defined as follows:

$$\text{CKA}(X, Y) = \frac{\|Y^T X\|_{\text{F}}^2}{(\|X^T X\|_{\text{F}} \|Y^T Y\|_{\text{F}})}, \tag{4.2}$$

where X and Y correspond respectively to the matrix of the d-dimensional mean-pooled (excluding special token) subword representations at layer l of the n parallel source and target sentences.

In Figure 4.7, we show the CKA similarity of monolingual models, compared with bilingual models and random encoders, of multi-way parallel sentences (Conneau et al., 2018) for five languages pair: English to English′ (obtained by back-translation from French),

French, German, Russian, and Chinese. The monolingual en′ is trained on the same data as en but with different random seed and the bilingual en-en′ is trained on English data but with separate embeddings matrix as in Section 4.5.3. The rest of the bilingual MLM is trained with the Default setting. We only use random encoder for non-English sentences.

Figure 4.7 shows bilingual models have slightly higher similarity compared to monolingual models with random encoders serving as a lower bound. Despite the slightly lower similarity between monolingual models, it still explains the alignment performance in Section 4.6.1. Because the measurement is also invariant to orthogonal mapping, the CKA similarity is highly correlated with the sentence-level alignment performance in Figure 4.6 with over 0.9 Pearson correlation for all four languages pairs. For monolingual and bilingual models, the first few layers have the highest similarity, which explains why Section 3.5 finds freezing bottom layers of mBERT helps cross-lingual transfer. On the other hand, the final few layers have the lowest similarity, perhaps because the model needs language specific information to solve the cloze task, which explains why Section 3.6 finds that mBERT retains strong language specific information. The similarity gap between monolingual model and bilingual model decreases as the languages pair become more distant. In other words, when languages are similar, using the same model increases representation similarity. On the other hand, when languages are dissimilar, using the same model does not help representation similarity much.

4.7 Discussion

In this chapter, we show that multilingual representations can emerge from unsupervised multilingual masked language models with only parameter sharing of some Transformer layers. Even without any anchor points, the model can still learn to map representations coming from different languages in a single shared embedding space. We also show that isomorphic embedding spaces emerge from monolingual masked language models in different languages, similar to word2vec embedding spaces (Mikolov, Le, and Sutskever, 2013). By using a linear mapping, we are able to align the embedding layers and the contextual representations of Transformers trained in different languages. We also use the CKA neural network similarity index to probe the similarity between BERT Models and show that the early layers of the Transformers are more similar across languages than the last layers. All of these effects were stronger for more closely related languages, suggesting there is room for significant improvements on more distant language pairs.

This type of emergent language universality has interesting theoretical and practical implications. We gain insight into why the models learn cross-lingual representation and open up new lines of inquiry into the implication of such emerge universality. It should be possible to adapt multilingual pretrained models to new languages with little additional training. For example, pretrained multilingual MLM models can be rapidly fine-tuned to another language (Artetxe, Ruder, and Yogatama, 2020; Chau, Lin, and Smith, 2020; Wang et al., 2020a; Pfeiffer et al., 2020).

Chapter 5

Are All Languages Created Equal in mBERT?

5.1 Introduction

In chapter 3, we show that mBERT learns high-quality cross-lingual representation and has strong zero-shot cross-lingual transfer performance. However, evaluations have focused on high resource languages, with cross-lingual transfer using English as a source language or within language performance. As chapter 3 evaluates mBERT on 39 languages, this leaves the majority of mBERT's 104 languages, most of which are low resource languages, untested.

In this chapter, we ask the following question. *Does mBERT learn equally high-quality representation for its 104 languages?* If not, which languages are hurt by its massively multilingual style pretraining? While it has been observed that for high resource languages like English, mBERT performs worse than monolingual BERT on English with the same capacity (Devlin, 2018). It is unclear that for low resource languages (in terms of monolingual corpus size), how does mBERT compare to a monolingual BERT? And, does multilingual joint training help mBERT learn better representation for low resource languages?

To answer this question, we first evaluate the representation quality of mBERT on 99 languages for NER, and 54 for part-of-speech tagging and dependency parsing. We show mBERT does not have equally high-quality representation for all of the 104 languages, with the bottom 30% languages performing much worse than a non-BERT model on NER. Additionally, by training various monolingual BERT for low-resource languages with the same data size, we show the low representation quality of low-resource languages is not the result of the hyperparameters of BERT or sharing the model with a large number of languages, as

monolingual BERT performs worse than mBERT. On the contrary, by pairing low-resource languages with linguistically-related languages, we show low-resource languages benefit from multilingual joint training, as bilingual BERT outperforms monolingual BERT while still lacking behind mBERT.

These experiments suggest that mBERT try its best to learn representation for the low resource languages with the given data. However, as BERT pretraining objective is not known for sample efficient, the small Wikipedia of low resource languages is not enough for mBERT to learn high quality representation. To address this challenge, we either need a more sample efficient pretraining algorithm or collect more data to make low resource languages high resource.

5.2 Background

Several factors need to be considered in understanding mBERT. First, the 104 most common Wikipedia languages vary considerably in size (Table 5.1). Therefore, mBERT training attempted to equalize languages by up-sampling sentences from low resource languages and down-sampling sentences from high resource languages. Second, while each language may be similarly represented in the training data, subwords are not evenly distributed among the languages. Many languages share common characters and cognates, biasing subword learning to some languages over others. Both of these factors may influence how well mBERT learns representations for low resource languages. Finally, Baevski et al. (2019) show that

in general larger pretraining data for English leads to better downstream performance, yet increasing the size of pretraining data exponentially only increases downstream performance linearly. For a low resource language with limited pretraining data, it is unclear whether contextual representations outperform previous methods.

5.2.1 Representations for Low Resource Languages

Embeddings with subword information, a non-contextual representation, like fastText (Bojanowski et al., 2017) and BPEmb (Heinzerling and Strube, 2018) are more data-efficient compared to contextual representation like ELMo and BERT when a limited amount of text is available. For low resource languages, there are usually limits on **monolingual corpora** and **task specific supervision**. When task-specific supervision is limited, e.g. sequence labeling in low resource languages, mBERT performs better than fastText while underperforming a single BPEmb trained on all languages (Heinzerling and Strube, 2019). Contrary to this work, we focus on mBERT from the perspective of representation learning for each language in terms of monolingual corpora resources and analyze how to improve BERT for low resource languages. We also consider parsing in addition to sequence labeling tasks.

WikiSize	Languages	# Languages	Size Range (GB)
3	io, pms, scn, **yo**	4	[0.006, 0.011]
4	cv, lmo, mg, min, su, vo	6	[0.011, 0.022]
5	an, bar, br, ce, fy, ga, gu, is, jv, ky, lb, **mn**, my, nds, ne, pa, pnb, sw, tg	19	[0.022, 0.044]
6	**af**, ba, cy, kn, la, mr, oc, sco, sq, tl, tt, uz	12	[0.044, 0.088]
7	az, bn, bs, eu, hi, ka, kk, lt, **lv**, mk, ml, nn, ta, te, ur	15	[0.088, 0.177]
8	ast, be, bg, da, el, et, gl, hr, hy, ms, sh, sk, sl, th, war	15	[0.177, 0.354]
9	fa, fi, he, id, ko, no, ro, sr, tr, vi	10	[0.354, 0.707]
10	ar, ca, cs, hu, nl, sv, uk	7	[0.707, 1.414]
11	ceb, it, ja, pl, pt, zh	6	[1.414, 2.828]
12	de, es, fr, ru	4	[2.828, 5.657]
14	en	1	[11.314, 22.627]

Table 5.1: List of 99 languages we consider in mBERT and its pretraining corpus size. Languages in **bold** are the languages we consider in Section 5.5.

5.3 Experiments

We begin by defining high and low resource languages in mBERT, a description of the models and downstream tasks we use for evaluation, followed by a description of the masked language model pretraining.

5.3.1 High/Low Resource Languages

Since mBERT was trained on articles from Wikipedia, a language is considered a high or low resource for mBERT based on the size of Wikipedia in that language. Size can be measured in many ways (articles, tokens, characters); we use the size of the raw dump archive file;[1] for convenience we use $\log_2$ of the size in MB (**WikiSize**). English is the highest resource language (15.5GB) and Yoruba the lowest (10MB).[2] Table 5.1 shows

[1]The size of English (en) is the size of this file: `https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2`

[2]The ordering does not necessarily match the number of speakers for a language.

languages and their relative resources.

5.3.2 Downstream Tasks

mBERT supports 104 languages, and we seek to evaluate the learned representations for as many of these as possible. We consider three NLP tasks for which annotated task data exists in a large number of languages: named entity recognition (NER), universal part-of-speech (POS) tagging and universal dependency parsing. For each task, we fine-tune a task-specific model built on top of the mBERT using within-language supervised data.

For NER we use data created by Pan et al. (2017), built automatically from Wikipedia, which covers 99 of the 104 languages supported by mBERT. We evaluate NER with entity-level F1. This data is in-domain as mBERT is pretrained on Wikipedia. For POS tagging and dependency parsing, we use Universal Dependencies (UD) v2.3 (Nivre, 2018b), which covers 54 languages (101 treebanks) supported by mBERT. We evaluate POS with accuracy (ACC) and Parsing with label attachment score (LAS) and unlabeled attachment score (UAS). For POS, we consider UPOS within the treebank. For parsing, we only consider universal dependency labels. The domain is treebank-specific so we use all treebanks of a language for completeness.

5.3.2.1 Task Models

For sequence labeling tasks (NER and POS), we add a linear function with a softmax on top of mBERT. For NER, at test time, we adopt the same post-processing step as

Section 3.2.3. For dependency parsing, we replace the LSTM in the graph-based parser of Dozat and Manning (2017) with mBERT. For the parser, we use the original hyperparameters. Note we do not use universal part-of-speech tags as input for dependency parsing. We fine-tune all parameters of mBERT for a specific task. We use a maximum sequence length of 128 for sequence labeling tasks. For sentences longer than 128, we use a sliding window with 64 previous tokens as context. For dependency parsing, we use sequence length 128 due to memory constraints and drop sentences with more than 128 subwords. We also adopt the same treatment for the baseline (Che et al., 2018) to obtain comparable results. Since mBERT operates on the subword-level, we select the first subword of each word for the task-specific layer with masking.

5.3.2.2 Task Optimization

We train all models with Adam (Kingma and Ba, 2014). We warm up the learning rate linearly in the first 10% steps then decrease linearly to 0. We select the hyperparameters based on dev set performance by grid search, as recommended by Devlin et al. (2019). The search includes a learning rate (2e-5, 3e-5, and 5e-5), batch size (16 and 32). As task-specific supervision size differs by language or treebank, we fine-tune the model for 10k gradient steps and evaluate the model every 200 steps. We select the best model and hyperparameters for a language or treebank by the corresponding dev set.

5.3.2.3 Task Baselines

We compare our mBERT models with previously published methods: Pan et al. (2017) for NER; For POS and dependency parsing the best performing system ranked by LAS in the 2018 universal parsing shared task (Che et al., 2018) [3], which use ELMo as well as word embeddings. Additionally, Che et al. (2018) is trained on POS and dependency parsing jointly while we trained mBERT to perform each task separately. As a result, the dependency parsing with mBERT does not have access to POS tags. By comparing mBERT to these baselines, we control for task and language-specific supervised training set size.

5.3.3 Masked Language Model Pretraining

We include several experiments in which we pretrain BERT from scratch. We use the PyTorch (Paszke et al., 2019) implementation by Conneau and Lample (2019), the same as Section 4.3. All sentences in the corpus are concatenated. For each language, we sample a batch of N sequence and each sequence contains M tokens, ignoring sentence boundaries. When considering two languages, we sample each language uniformly. We then randomly select 15% of the input tokens for masking, proportionally to the exponentiated token count of power -0.5, favoring rare tokens. We replace selected masked token with <MASK> 80% of the time, the original token 10% of the time, and uniform random token within the vocabulary 10% of the time. The model is trained to recover the original token (Devlin

[3]The shared task uses UD v2.2 while we use v2.3. However, treebanks contain minor changes from version to version.

et al., 2019). We drop the next sentence prediction task as Liu et al. (2019b) find it does not improve downstream performance.

5.3.3.1 Data Processing

We extract text from a Wikipedia dump with Gensim (Řehůřek and Sojka, 2010). We learn vocabulary for the corpus using SentencePiece (Kudo and Richardson, 2018) with the unigram language model (Kudo, 2018). When considering two languages, we concatenate the corpora for the two languages while sampling the same number of sentences from both corpora when learning vocabulary. We learn a vocabulary of size V, excluding special tokens. Finally, we tokenized the corpora using the learned SentencePiece model and did not apply any further preprocessing.

5.3.3.2 BERT Models

Following mBERT, We use 12 Transformer layers (Vaswani et al., 2017) with 12 heads, embedding dimensions of 768, hidden dimension of the feed-forward layer of 3072, dropout of 0.1 and GELU activation (Hendrycks and Gimpel, 2016). We tie the output softmax layer and input embeddings (Press and Wolf, 2017). We consider both a 12 layer model (**base**) and a smaller 6 layer model (**small**).

5.3.3.3 BERT Optimization

We train BERT with Adam and an inverse square root learning rate scheduler with warmup (Vaswani et al., 2017). We warm up linearly for 10k steps and the learning rate is 0.0001. We use batch size $N = 88$ and mixed-precision training. We trained the model for roughly 115k steps and save a checkpoint every 23k steps, which corresponds to 10 epochs. We select the best out of five checkpoints with a task-specific dev set. We train each model on a single NVIDIA RTX Titan with 24GB of memory for roughly 20 hours.

5.4 Are All Languages Created Equal in mBERT?

Figure 5.1 shows the performance of mBERT and the baseline averaged across all languages by Wikipedia size (see Table 5.1 for groupings). For WikiSize over 6, mBERT is comparable or better than baselines in all three tasks, with the exception of NER. For NER in very high resource languages (WikiSize over 11, i.e. top 10%) mBERT performs worse than baseline, suggesting high resource languages could benefit from monolingual pretraining. Note mBERT has strong UAS on parsing but weak LAS compared to the baseline; in Section 3.4 we find adding POS to mBERT improves LAS significantly. We expect multitask learning on POS and parsing could further improve LAS. While POS and Parsing only cover half (54) of the languages, NER covers 99 of 104 languages, extending the curve to the lowest resource languages. mBERT performance drops significantly for languages with WikiSize less than 6 (bottom 30% languages). For the smallest size, mBERT

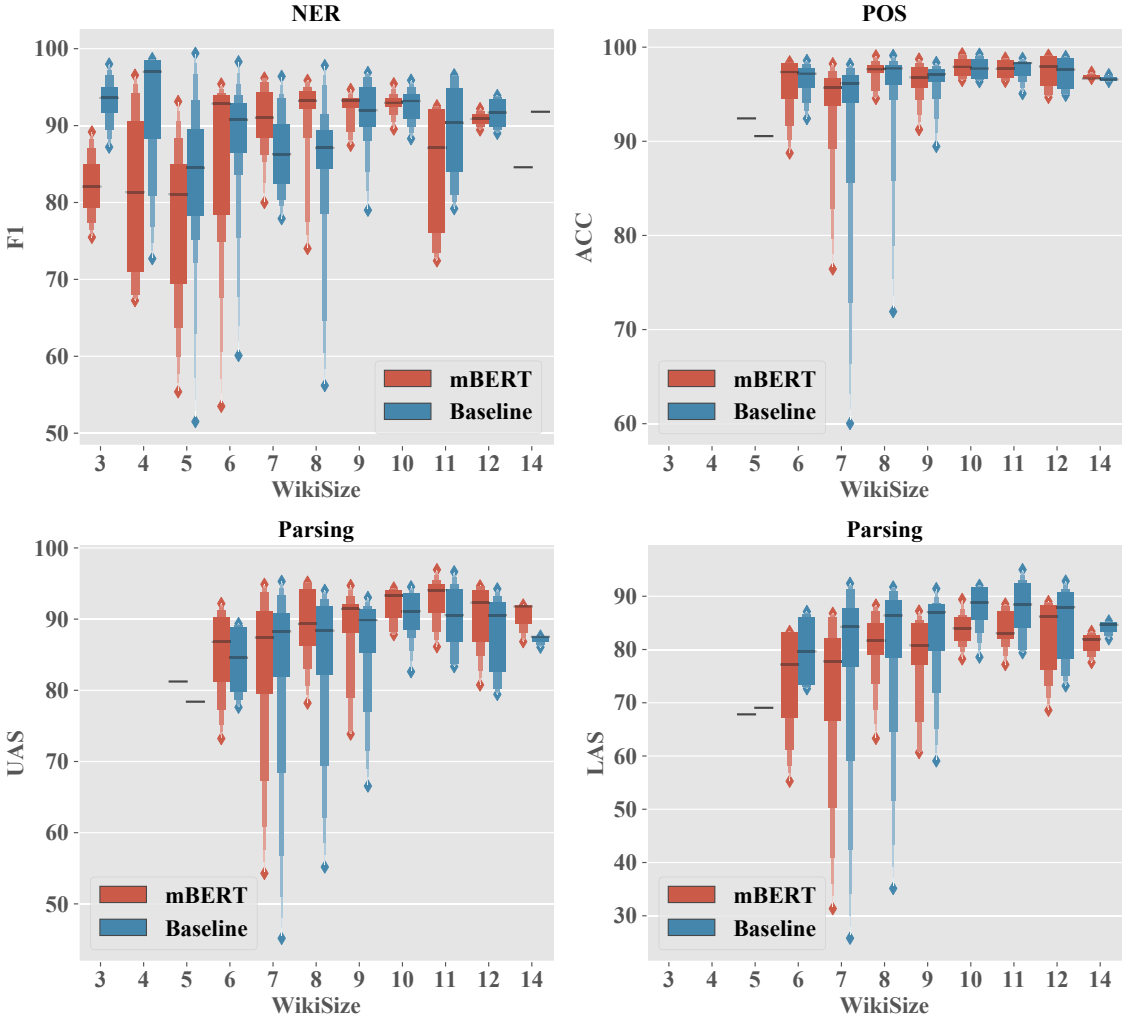


Figure 5.1: mBERT vs baseline grouped by WikiSize. mBERT performance drops much more than baseline models on languages lower than WikiSize 6 – the bottom 30% languages supported by mBERT – especially in NER, which covers nearly all mBERT supported languages.

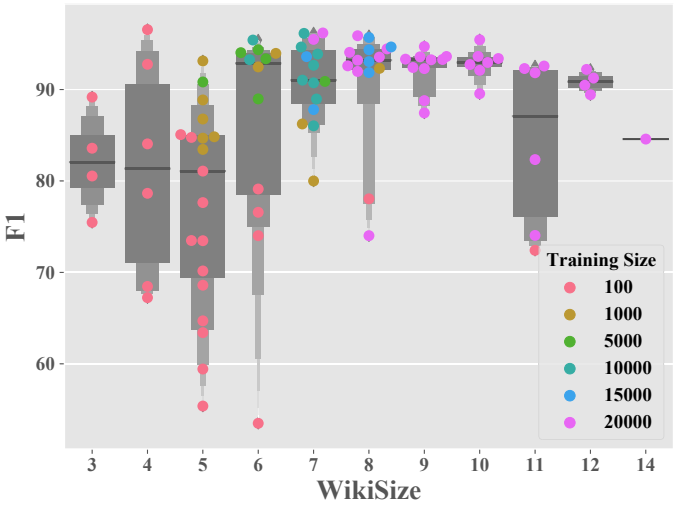


Figure 5.2: NER with mBERT on 99 languages, ordered by size of pretraining corpus (WikiSize). Task-specific supervised training size differs by language. Performance drops dramatically with less pretraining and supervised training data.

goes from being competitive with state-of-the-art to being *over 10 points behind.* Readers may find this surprising since while these are very low resource languages, mBERT training up-weighted these languages to counter this effect.

Figure 5.2 shows the performance of mBERT (only) for NER over languages with *different resources*, where we show how much task-specific supervised training data was available for each language. For languages with only 100 labeled sentences, the performance of mBERT drops significantly as these languages also had less pretraining data. While we may expect that pretraining representations with mBERT would be most beneficial for languages with only 100 labels, as Howard and Ruder (2018) show pretraining improve data-efficiency for English on text classification, our results show that on low resource languages this strategy performs much worse than a model trained directly on the available task data. Clearly, mBERT provides variable quality representations depending on the

	Coefficient	p-value	CI
Univariate			
Training Size	0.035	<0.001	[0.029, 0.041]
Training Vocab	0.021	<0.001	[0.017, 0.025]
WikiSize	0.015	<0.001	[0.007, 0.023]
Multivariate			
Training Size	0.029	<0.001	[0.023, 0.035]
WikiSize	-0.014	<0.001	[-0.022, -0.006]

Table 5.2: Statistical analysis on what factors predict downstream performance. We fit two types of linear models, which consider either single factor or multiple factors.

language. While we confirm the finding of others that mBERT is excellent for high resource languages, it is much worse for low resource languages. Our results suggest caution for those expecting a reliable model for *all* 104 mBERT languages.

5.5 Why Are All Languages Not Created Equal in mBERT?

5.5.1 Statistical Analysis

We present a statistical analysis to understand why mBERT does so poorly in some languages. We consider three factors that might affect the downstream task performance: pretraining Wikipedia size (WikiSize), task-specific supervision size, and vocabulary size in task-specific data. Note we take $\log_2$ of training size and training vocab following WikiSize.

We consider NER because it covers nearly all languages of mBERT.

We fit a linear model to predict task performance (F1) using a single factor. Table 5.2 shows that each factor has a statistically significant positive correlation. One unit increase of training size leads to the biggest performance increase, then training vocabulary followed by WikiSize, all in log scale. Intuitively, training size and training vocab correlate with each other. We confirm this with a log-likelihood ratio test; adding training vocabulary to a linear model with training size yields a statistically insignificant improvement. As a result, when considering multiple factors, we consider training size and WikiSize. Interestingly, Table 5.2 shows training size still has a positive but slightly smaller slope, but the slope of WikiSize change sign, which suggests WikiSize might correlate with training size. We confirm this by fitting a linear model with training size as x and WikiSize as y and the slope is over 0.5 with $p < 0.001$. This finding is unsurprising as the NER dataset is built from Wikipedia so larger Wikipedia size means larger training size.

In conclusion, the larger the task-specific supervised dataset, the better the downstream performance on NER. Unsurprisingly, while pretraining improve data-efficiency (Howard and Ruder, 2018), it still cannot solve a task with limited supervision. Training vocabulary and Wikipedia size correlate with training size, and increasing either one factor leads to better performance. A similar conclusion could be found when we try to predict the performance ratio of mBERT and the baseline instead. Statistical analysis shows a correlation between resource and mBERT performance but can not give a causal answer on why low resource languages within mBERT perform poorly.

	lv	af	mn	yo
Genus	Baltic	Germanic	Mongolic	Defoid
Family	Indo-Eur	Indo-Eur	Altaic	Niger-Congo
WikiSize	7	6	5	3
# Sentences (M)	2.9	2.3	0.8	0.1
# Tokens (M)	21.8	28.8	6.4	0.9
mBERT vocab (K)	56.6	59.0	42.3	29.3
mBERT vocab (%)	49.2	51.3	36.8	25.5

Table 5.3: Statistic of four low resource languages.

5.5.2 mBERT vs monolingual BERT

We have established that mBERT does not perform well in low-resource languages. Is this because we are relying on a multilingual model that favors high-resource over low-resource languages? To answer this question we train monolingual BERT models on several low resource languages with different hyperparameters. Since pretraining a BERT model from scratch is computationally intensive, we select four low resource languages: Latvian (lv), Afrikaans (af), Mongolian (mn), and Yoruba (yo). These four languages (bold font in Table 5.3) reflect varying amounts of monolingual training data.

It turns out that these low resource languages are reasonably covered by mBERT's vocabulary: 25% to 50% of the subword types within the mBERT 115K vocabulary appear in these languages' Wikipedia. However, the mBERT vocabulary is by no means optimal for these languages. Figure 5.3 shows that a large amount of the mBERT vocabulary that appears in these languages is low frequency while the language-specific SentencePiece vocabulary has a much higher frequency. In other words, the vocabulary of mBERT is not

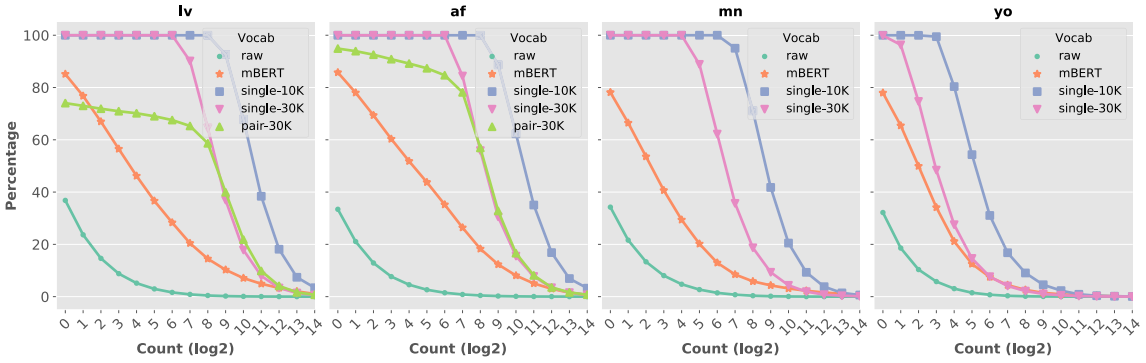


Figure 5.3: Percentage of vocabulary containing word count larger than a threshold. "Raw" is the vocabulary segmented by space. Single-30K and Single-10K are 30K/10K vocabularies learned from single languages. Pair-30K is 30K vocabulary learned from the selected language and a closely related language, described in Section 5.5.3.

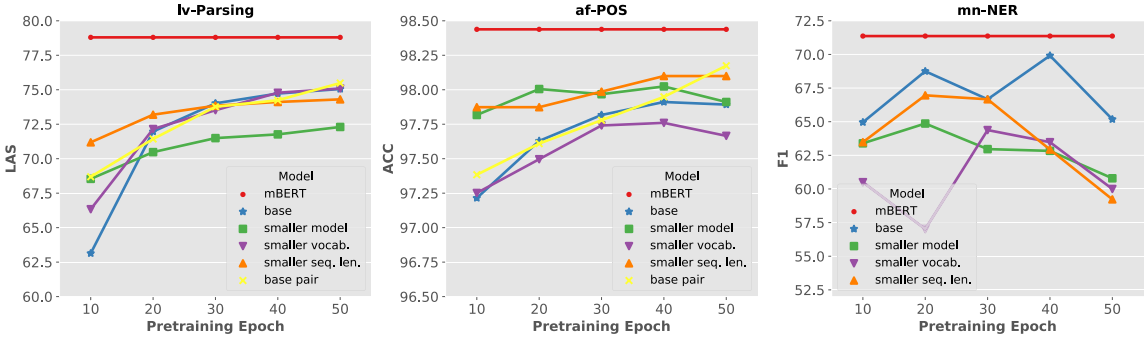


Figure 5.4: Dev performance with different pretraining epochs on three languages and tasks. Dev performance on higher resources languages (lv, af) improves as training continues, while lower resource languages (mn) fluctuate.

distributed uniformly.

To train the monolingual BERTs properly for low resource languages, we consider four different sets of hyperparameters. In **base**, we follow English monolingual BERT on learning vocabulary size $V = 30K$, 12 layers of transformer (base). To ensure we have a reasonable batch size for training using our GPU, we set the training sequence length to $M = 256$. Since a smaller model can prevent overfitting smaller datasets, we

Model Size	Vocabulary	Max Length	lv NER	lv POS	lv Parsing (LAS/UAS)	af NER	af POS	af Parsing (LAS/UAS)	mn NER	yo NER
Baseline										
Baseline			92.10	**96.19**	**84.47**/88.28	**94.00**	97.50	**85.69**/88.67	**76.40**	**94.00**
mBERT			**93.88**	95.69	77.78/**88.69**	93.36	**98.26**	83.18/**89.69**	64.71	80.54
Monolingual BERT (Section 5.5.2)										
base	30k	256	93.02	95.76	74.18/85.35	90.90	97.76	80.08/86.92	56.20	72.57
small	-	-	92.75	95.41	71.67/83.34	90.67	98.02	80.60/87.40	58.92	70.80
-	10k	-	92.68	95.65	73.94/85.20	89.55	97.66	79.91/86.93	41.70	80.18
-	-	128	93.38	95.57	73.21/84.53	91.84	97.87	80.83/87.59	55.91	73.45
Bilingual BERT (Section 5.5.3)			lv + lt			af + nl				
base	30k	256	93.22	96.03	74.42/85.60	91.85	97.98	81.73/88.55	n/a	n/a

Table 5.4: Monolingual BERT on four languages with different hyperparameters. Underscore denotes best within monolingual BERT and **bold** denotes best among all models. Monolingual BERT underperforms mBERT in most cases. "-" denotes same as base case.

consider 6 transformer layers (**small**). We do not change the batch size as a larger batch is observed to improve performance (Liu et al., 2019b). As low resource languages have small corpora, 30K vocabulary items might not be optimal. We consider **smaller vocabulary** with $V = 10K$. Finally, since in fine-tuning we only use a maximum sequence length of 128, in **smaller sequence length**, we match the fine-tuning phrase with $M = 128$. As a benefit of half the self-attention range, we can increase the batch size over 2.5 times to $N = 220$.

Table 5.4 shows the performance of monolingual BERT in four settings. The model with smaller sequence length performs best for monolingual BERT and outperforms the base model in 5 out of 8 tasks and languages combination. The model with smaller vocabulary has mixed performance in the low resource languages (mn, yo) but falls short for (relatively) higher resource languages (lv, af). Finally, the smaller model underperforms the base model in 5 out of 8 cases. In conclusion, the best way to pretrain BERT with a limited amount of computation for low resource languages is to use a smaller sequence length to allow a larger

batch size.

Despite these insights, no monolingual BERT outperforms mBERT (except Latvian POS). For higher resource languages (lv, af) we hypothesize that training longer with larger batch size could further improve the downstream performance as the cloze task dev perplexity was still improving. Figure 5.4 supports this hypothesis showing downstream dev performance of lv and af improves as pretraining continues. Yet for lower resource languages (mn, yo), the cloze task dev perplexity is stuck and we began to overfit the training set. At the same time, Figure 5.4 shows the downstream performance of mn fluctuates. It suggests the cloze task dev perplexity correlates with downstream performance when dev perplexity is not decreasing.

The fact that monolingual BERT underperforms mBERT on four low resource languages suggests that mBERT style multilingual training benefits low resource languages by transferring from other languages; monolingual training produces worse representations due to small corpus size. Additionally, the poor performance of mBERT on low resource languages does not emerge from balancing between languages. Instead, it appears that we do not have sufficient data, or the model is not sufficiently data-efficient.

5.5.3 mBERT vs Bilingual BERT

Finally, we consider a middle ground between monolingual training and massively multilingual training. We train a BERT model on a low resource language (lv and af) paired with a related higher resource language. We pair Lithuanian (lt) with Latvian and Dutch

(nl) with Afrikaans.[4] Lithuanian has a similar size to Latvian while Dutch is over 10 times bigger. Lithuanian belongs to the same Genus as Latvian while Afrikaans is a daughter language of Dutch. The **base pair** model has the same hyperparameters as the base model.

Table 5.4 shows that pairing low resource languages with closely related languages improves downstream performance. The Afrikaans-Dutch BERT improves more compared to Latvian-Lithuanian, possibly because Dutch is much larger than Afrikaans, as compared to Latvian and Lithuanian. These experiments suggest that pairing linguistically related languages can benefit representation learning and adding extra languages can further improve the performance as demonstrated by mBERT. It echoes the finding of Conneau and Lample (2019) where multilingual training improves uni-directional language model perplexity for low resource languages. Concurrent to the publication of this chapter, Conneau et al. (2020a) shows similar findings as the performance of low resource languages (Urdu and Swahili) improves on XNLI when more languages are trained jointly then decrease with an increasing number of languages. However, they do not consider the effect of language similarity.

5.6 Discussion

While mBERT covers 104 languages, in this chapter, we find that the 30% languages with least pretraining resources perform worse than using no pretrained language model at all. Therefore, we caution against using mBERT alone for low resource languages. Furthermore, training a monolingual model on low resource languages does no better. Training on pairs of

[4]We did not consider mn and yo since neither has a closely related language in mBERT.

closely related low resource languages helps but still lags behind mBERT. Thus, mBERT is trying its best to learn representation for low resource languages. However, constrained by the sample inefficiency of BERT objective and the lack of data of low resource languages, mBERT learns low quality representation for low resource languages. On the other end of the spectrum, the highest resource languages (top 10%) are hurt by massively multilingual joint training. While mBERT has access to numerous languages, the resulting model is worse than a monolingual model when sufficient training data exists.

Our findings suggest, with small monolingual corpus, BERT does not learn high-quality representation for low resource languages. To learn better representation for low resource languages, we suggest either collect more data to make low resource language high resource, which leads to XLM-R (Conneau et al., 2020a), or consider more data-efficient pretraining techniques like Clark et al. (2020), which leads to better performing XLM-E (Chi et al., 2021b). On the other hand, for high resource languages, training a monolingual model is likely to produce better representation than mBERT. Since English BERT and multilingual BERT, a large number of BERT-like models for various languages have been publicly available, e.g. Dutch (Delobelle, Winters, and Berendt, 2020), French (Martin et al., 2020), and Vietnamese (Nguyen and Tuan Nguyen, 2020). In fact, by November 2021, over 1700 BERT-like models are available with the Transformers library (Wolf et al., 2020).

Chapter 6

How To Inject Cross-lingual Signals Into Multilingual Encoders?

6.1 Introduction

Massively multilingual encoders including multilingual BERT (Devlin et al., 2019, mBERT) and XLM-RoBERTa (Conneau et al., 2020a, XLM-R) are pretrained without any explicit cross-lingual signal. In this chapter, we will investigate how to inject two types of cross-lingual signal into multilingual encoders: bilingual dictionary and bitext.

Bilingual dictionary is widely available for most language pairs, and it is easy to collect bilingual dictionary for a new language pair (Kamholz, Pool, and Colowick, 2014). We inject it into the pretraining process by increasing subwords overlap across languages. We achieve additional subwords overlap by creating synthetic code-switching corpus with bilingual dictionary. As we observe in Section 3.7, subwords overlap between languages correlates with cross-lingual transfer performance, although Section 4.5.2 shows that subword overlap is not the necessary condition for cross-lingual representation. In Section 6.3, we show that the correlation indeed holds with additional subwords overlap, in other word, having extra anchor points benefit the cross-lingual representation.

Bitext is available for most high-resource language pairs (usually involving English), and researchers have proposed collecting additional bitext by mining parallel sentences from the Web (Schwenk et al., 2021; Schwenk et al., 2019). While bitext can be incorporated during expensive pretraining (Conneau and Lample, 2019; Huang et al., 2019; Ji et al., 2020; Chi et al., 2021a), aligning pretrained multilingual encoders with explicit alignment objective, i.e. enforcing similar words from different languages have similar representation, is much more efficient. However, as word-level alignments from an unsupervised aligner

are often suboptimal, in Section 6.4, we develop a new cross-lingual alignment objective for training our model. We base our objective on contrastive learning, in which two similar inputs – such as from a bitext – are directly optimized to be similar, relative to a negative set. These methods have been effective in computer vision tasks (He et al., 2020; Chen et al., 2020).

Most previous work on contextual alignments consider high-quality bitext like Europarl (Koehn, 2005) or MultiUN (Eisele and Chen, 2010). While helpful, these resources are unavailable for most languages for which we seek a zero-shot transfer. To better reflect the quality of bitext available for most languages, we additionally use OPUS-100 (Zhang et al., 2020), a randomly sampled 1 million subset (per language pair) of the OPUS collection (Tiedemann, 2012). In Section 6.4, we show that our new contrastive learning alignment objectives outperform previous work (Cao, Kitaev, and Klein, 2020) when applied to bitext from previous works or the OPUS-100 bitext. However, our experiments also produce a negative result. While previous work showed improvements from alignment-based objectives on zero-shot cross-lingual transfer for a single task (XNLI) with a single random seed, our more extensive analysis tells a different story. We report the mean and standard deviation of multiple runs with the same hyperparameters and different random seeds. We find that previously reported improvements disappear, even while our new method shows a small improvement. Furthermore, we extend the evaluation to multiple languages on 4 tasks, further supporting our conclusions.

6.2 Background

6.2.1 Explicit Alignment Objectives

We begin with a presentation of explicit alignment objective functions that use parallel data across languages for training multilingual encoders. These objectives assume multilingual data in the form of word pairs in parallel sentences. Since gold word alignments are scarce, we use an unsupervised word aligner. Let $\mathbf{S}$ and $\mathbf{T}$ be the contextual hidden state matrix of corresponding words from a pretrained multilingual encoder. We assume $\mathbf{S}$ is English while $\mathbf{T}$ is a combination of different target languages. As both mBERT and XLM-R operate at the subword level, we use the representation of the first subword, which is consistent with the evaluation stage. Each s_i and t_i are a corresponding row of $\mathbf{S}$ and $\mathbf{T}$, respectively. $\mathbf{S}$ and $\mathbf{T}$ come from the final layer of the encoder while $\mathbf{S}^l$ and $\mathbf{T}^l$ come from the l^{th}-layer.

6.2.1.1 Linear Mapping

If $\mathbf{S}$ and $\mathbf{T}$ are static feature (such as from ELMo (Peters et al., 2018)) then $\mathbf{T}$ can be aligned so that it is close to $\mathbf{S}$ via a linear mapping (Wang et al., 2019; Wang et al., 2020b; Liu et al., 2019a; Conneau et al., 2020b), similar to aligning monolingual embeddings to produce cross-lingual embeddings. For feature $\mathbf{S}^l$ and $\mathbf{T}^l$ from layer l, we can learn a

mapping $\mathbf{W}^l$.

$$\mathbf{W}^{l*} = \arg\min_{\mathbf{W}^l} \|\mathbf{S}^l - \mathbf{T}^l\mathbf{W}^l\|_2^2 \tag{6.1}$$

When $\mathbf{W}^l$ is orthogonal, Equation 6.1 is known as Procrustes problem (Smith et al., 2017) and can be solved by SVD. Alternatively, Equation 6.1 can also be solved by gradient descent, without the need to store in memory huge matrices $\mathbf{S}$ and $\mathbf{T}$. We adopt the latter more memory efficient approach. Following Lample et al. (2018), we enforce the orthogonality by alternating the gradient update and the following update rule

$$\mathbf{W} \leftarrow (1+\beta)\mathbf{W} - \beta(\mathbf{W}\mathbf{W}^T)\mathbf{W} \tag{6.2}$$

with $\beta = 0.01$. Note we learn different $\mathbf{W}^l$ for each target language.

This approach has yielded improvements in several studies. In Section 4.6.1.2, we use bilingual BERT and 10k parallel sentences from XNLI (Conneau et al., 2018) to improve dependency parsing (but not NER) on French, Russian, and Chinese. Wang et al. (2019) use mBERT and 10k parallel sentences from Europarl to improve dependency parsing. Wang et al. (2020b) use mBERT and 30k parallel sentences from Europarl to improve named entity recognition (NER) on Spanish, Dutch, and German. Liu et al. (2019a) do not evaluate on cross-lingual transfer tasks.

6.2.1.2 L2 Alignment

Instead of using $\mathbf{S}$ and $\mathbf{T}$ as static features, Cao, Kitaev, and Klein (2020) propose fine-tuning the entire encoder

$$\mathcal{L}_{\text{L2}}(\theta) = \text{mean}_i(\|s_i - t_i\|_2^2) \tag{6.3}$$

where θ is the encoder parameters. To prevent a degenerative solution, they additionally use a regularization term

$$\mathcal{L}_{\text{reg-hidden}}(\theta) = \|\bar{\mathbf{S}} - \bar{\mathbf{S}}_{\text{pretrained}}\|_2^2 \tag{6.4}$$

where $\bar{\mathbf{S}}$ denote **all** hidden states of the source sentence including unaligned words, encouraging the source hidden states to stay close to the pretrained hidden states. With mBERT and 20k to 250k parallel sentences from Europarl and MultiUN, Cao, Kitaev, and Klein show improvement on XNLI but not parsing.[1]

In preliminary experiments, we found constraining parameters to stay close to their original pretrained values also prevents degenerative solutions

$$\mathcal{L}_{\text{reg-param}}(\theta) = \|\theta - \theta_{\text{pretrained}}\|_2^2 \tag{6.5}$$

while being more efficient than Equation 6.4. As a result, we adopt the following objective

[1]The authors state they did not observe improvements on parsing in the NLP Hightlights podcast (#112) (AI2, 2020).

(with $\lambda = 1$):

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{L2}}(\theta) + \lambda \mathcal{L}_{\text{reg-param}}(\theta) \tag{6.6}$$

6.3 Bilingual Dictionary

6.3.1 Experiments

As Section 3.7 suggests that there may be correlation between cross-lingual performance and anchor points, we additionally increase anchor points by using bilingual dictionary to create code switch data for training bilingual MLM. Specifically, for two languages, ℓ_1 and ℓ_2, with bilingual dictionary entries d_{ℓ_1,ℓ_2}, we add anchors to the training data as follows. For each training word w_{ℓ_1} in the bilingual dictionary, we either leave it as is (70% of the time) or randomly replace it with one of the possible translations from the dictionary (30% of the time). We change at most 15% of the words in a batch and sample word translations from PanLex (Kamholz, Pool, and Colowick, 2014) bilingual dictionary, weighted according to their translation quality.[2] We pretrain two bilingual encoders for each language pair: with or without synthetic code-switching corpus. We consider the same three language pairs as Section 4.5: English-French, English-Russian, and English-Chinese. The rest of the pretraining is the same as Section 4.3. Recall that each encoder is a 8-layer Transformer. To ensure a fair comparison, both models have the same number of gradient updates. For

[2]Although we only consider pairs of languages, this procedure naturally scales to multiple languages.

	XNLI (Acc)			NER (F1)			Parsing (LAS)		
	fr	ru	zh	fr	ru	zh	fr	ru	zh
Default	73.6	68.7	68.3	**79.8**	**60.9**	63.6	73.2	56.6	28.8
+ Bi. Dict.	**74.0**	**69.8**	**72.1**	76.1	59.7	**66.8**	**73.3**	**56.9**	**29.2**

Table 6.1: Impact of extra anchor points with synthetic code-switching corpus based on bilingual dictionary.

this section, we adapt the same zero-shot cross-lingual evaluation on XNLI, NER, and dependency parsing as Section 4.4

6.3.2 Fingdings

Table 6.1 shows using bilingual dictionary to create synthetic code-switching corpus overall benefit cross-lingual representation. Anchor points have a clear effect on performance and more anchor points help, especially in the less closely related language pairs (e.g. English-Chinese has a larger effect than English-French with over 3 points improvement on NER and XNLI).

6.4 Bitext

6.4.1 Contrastive Alignment

Inspired by the contrastive learning framework of Chen et al. (2020), we propose a contrastive loss to align $\mathbf{S}$ and $\mathbf{T}$ by fine-tuning the encoder. Assume in each batch, we have

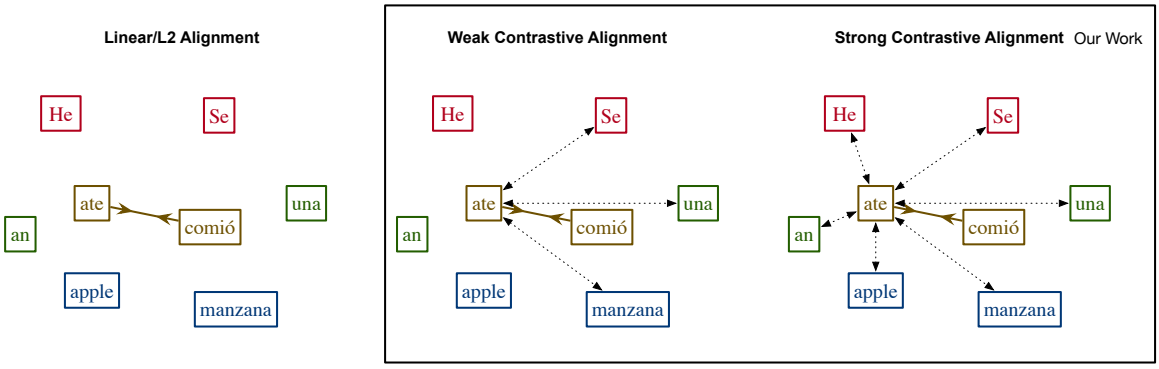


Figure 6.1: Explicit alignment with different objectives. We use a parallel sentence "He ate an apple" and "Se comió una manzana" as an example. While linear or L2 alignment optimizes for absolute distance, making "ate" and "comió" as close as possible (solid line), contrastive alignment optimizes for relative distance, making "ate" and "comió" closer (solid line) and pushing other away (dotted line).

corresponding (s_i, t_i) where $i \in \{1, \ldots, B\}$. Instead of optimizing the absolute distance between s_i and t_i like Equation 6.1 or Equation 6.3 in Section 6.2.1, contrastive loss allows more flexibility by encouraging s_i and t_i to be closer as compared with any other hidden state. In other words, our proposed contrastive alignment optimizes the relative distance between s_i and t_i (see Figure 6.1 for visualization). As the alignment signal is often suboptimal, our alignment objective is more robust to errors in unsupervised word-level alignment. Additionally, unlike previous works, we select different sets of negative examples to enforce different levels of cross-lingual alignment. Finally, it naturally scales to multiple languages.

6.4.1.1 Weak alignment

When the negative examples only come from target languages, we enforce a weak cross-lingual alignment, i.e. s_i should be closer to t_i than any other $t_j, \forall j \neq i$. The same is

true in the other direction. The loss of a batch is

$$\mathcal{L}_{\text{weak}}(\theta) = \frac{1}{2B}\sum_{i=1}^{B}(\log \frac{\exp(\text{sim}(s_i, t_i)/T)}{\sum_{j=1}^{B}\exp(\text{sim}(s_i, t_j)/T)} + \log \frac{\exp(\text{sim}(s_i, t_i)/T)}{\sum_{j=1}^{B}\exp(\text{sim}(s_j, t_i)/T)}) \quad (6.7)$$

where $T = 0.1$ is a temperature hyperparameter and $\text{sim}(a, b)$ measures the similarity of a and b.

We use a learned cosine similarity $\text{sim}(a, b) = \cos(f(a), f(b))$ where f is a feed-forward feature extractor with one hidden layer (768-768-128) and ReLU. It can learn to discard language-specific information and only align the align-able information. Chen et al. (2020) find that this similarity measure learns better representation for computer vision. After alignment, f is discarded as most cross-lingual transfer tasks do not need this feature extractor, though tasks like parallel sentence retrieval might find it helpful. This learned similarity cannot be applied to an absolute distance objective like Equation 6.3 as it can produce degenerate solutions.

6.4.1.2 Strong alignment

If the negative examples include both source and target languages, we enforce a strong cross-lingual alignment, i.e. s_i should be closer to t_i than any other $t_j, \forall j \neq i$ and $s_j, \forall j \neq i$.

$$\mathcal{L}_{\text{strong}}(\theta) = \frac{1}{2B}\sum_{h\in\mathcal{H}}\log \frac{\exp(\text{sim}(h, \text{aligned}(h))/T)}{\sum_{h'\in\mathcal{H}, h'\neq h}\exp(\text{sim}(h, h')/T)} \quad (6.8)$$

where $\text{aligned}(h)$ is the aligned hidden state of h and $\mathcal{H} = \{s_1, \ldots, s_B, t_1, \ldots, t_B\}$.

For both weak and strong alignment objectives, we add a regularization term Equation 6.5 with $\lambda = 1$.

6.4.2 Experiments

6.4.2.1 Multilingual Alignment

We consider alignment and transfer from English to 8 target languages: Arabic, German, English, Spanish, French, Hindi, Russian, Vietnamese, and Chinese. We use two sets of bitexts: (1) bitext used in previous works (Conneau and Lample, 2019) and (2) the OPUS-100 bitext (Zhang et al., 2020). (1) For bitext used in previous works, we use MultiUN for Arabic, Spanish, French, Russian or Chinese, EUBookshop (Skadiņš et al., 2014) for German, IIT Bombay corpus (Kunchukuttan, Mehta, and Bhattacharyya, 2018) for Hindi and OpenSubtitles (Lison, Tiedemann, and Kouylekov, 2018) for Vietnamese. We sample 1M bitext for each target language. (2) The OPUS-100 covers 100 languages with English as the center, and sampled from the OPUS collection randomly, which better reflects the average quality of bitext for most languages. It contains 1M bitext for each target language, except Hindi (0.5M).

We tokenize the bitext with Moses (Koehn et al., 2007) and segment Chinese with Chang, Galley, and Manning (2008). We use `fast_align` (Dyer, Chahuneau, and Smith, 2013) to produce unsupervised word alignments in both directions and symmetrize with the *grow-diag-final-and* heuristic. We only keep one-to-one alignment and discard any trivial

alignment where the source and target words are identical.

We train the L2 (Section 6.2.1.2), weak, and strong alignment objectives in a multilingual fashion. Each batch contains examples from all target languages. Following Devlin et al. (2019), we optimize with Adam (Kingma and Ba, 2014), learning rate `1e-4`, 128 batch size, 100k total steps ($\approx$ 2 epochs), 4k steps linear warmup and linear decay. We use 16-bit precision and train each model on a single RTX TITAN for around 18 hours. We set the maximum sequence length to 96. For linear mapping (Section 6.2.1.1), we use a linear decay learning rate from `1e-4` to 0 in 20k steps ($\approx$ 3 epochs), and train for 3 hours for each language pairs.

6.4.2.2 Evaluation

We consider zero-shot cross-lingual transfer with XNLI (Conneau et al., 2018), NER (Pan et al., 2017), POS tagging and dependency parsing (Zeman, 2020a).[3] We evaluate XNLI and POS tagging with accuracy (ACC), NER with span-level F1, and parsing with labeled attachment score (LAS). For the task-specific layer, we use a linear classifier for XNLI, NER, and POS tagging, and use Dozat and Manning (2017) for dependency parsing. We fine-tune all parameters on English training data and directly transfer to target languages. We optimize with Adam, learning rate `2e-5` with 10% steps linear warmup and linear decay, 5 epochs, and 32 batch size. For the linear mapping alignment, we use an ELMo-style

[3]We use the following treebanks: Arabic-PADT, German-GSD, English-EWT, Spanish-GSD, French-GSD, Hindi-HDTB, Russian-GSD, Vietnamese-VTB, and Chinese-GSD.

feature-based model[4] with 4 extra Transformer layers (Vaswani et al., 2017), a CRF instead of a linear classifier for NER, and train for 20 epochs, a batch size of 128 and learning rate `1e-3` (except NER and XNLI with `1e-4`). All token level tasks use the first subword as the word representation for task-specific layers similar to previous chapters. Model selection is done on the English dev set. We report the mean and standard derivation of test performance of 5 evaluation runs with different random seeds[5] and the same hyperparameters.

We set the maximum sequence length to 128 during fine-tuning. For NER and POS tagging, we additionally use a sliding window of context to include subwords beyond the first 128. At test time, we use the same maximum sequence length except for parsing. At test time for parsing, we only use the first 128 words of a sentence instead of subwords to make sure we compare different models consistently. We ignore words with POS tags of `SYM` and `PUNCT` during parsing evaluation. We adopt the same post-processing heuristic steps as Section 3.2.3 during NER evaluation. As the supervision on Chinese NER is on character-level, we segment the character into word using the Stanford Word Segmenter and realign the label.

	XNLI	NER	POS	Parsing
mBERT	$70.1_{\pm 0.8}$	$67.7_{\pm 1.3}$	$78.3_{\pm 0.5}$	$52.6_{\pm 0.4}$
+ Linear Mapping	$70.0_{\pm 0.6}$	$63.7_{\pm 1.5}$	$79.5_{\pm 0.5}$	$53.6_{\pm 0.3}$
+ L2 Align	$69.7_{\pm 0.4}$	$67.1_{\pm 1.0}$	$78.0_{\pm 1.3}$	$52.2_{\pm 0.7}$
+ Weak Align (Our)	$70.5_{\pm 0.7}$	$68.0_{\pm 1.3}$	$78.8_{\pm 0.7}$	$53.1_{\pm 0.6}$
+ Strong Align (Our)	$70.4_{\pm 0.7}$	$67.7_{\pm 1.1}$	$79.0_{\pm 0.7}$	$53.0_{\pm 0.6}$
XLM-R_{base}	$76.4_{\pm 0.5}$	$66.4_{\pm 0.9}$	$81.2_{\pm 0.6}$	$57.3_{\pm 0.6}$
+ Linear Mapping	$73.4_{\pm 0.6}$	$54.1_{\pm 0.9}$	$81.3_{\pm 0.5}$	$55.6_{\pm 0.5}$
+ L2 Align	$75.7_{\pm 0.5}$	$65.7_{\pm 1.2}$	$81.3_{\pm 0.9}$	$56.2_{\pm 0.7}$
+ Weak Align (Our)	$76.1_{\pm 0.7}$	$66.0_{\pm 1.0}$	$81.5_{\pm 0.5}$	$57.4_{\pm 0.4}$
+ Strong Align (Our)	$76.0_{\pm 0.6}$	$66.1_{\pm 0.9}$	$81.4_{\pm 0.6}$	$57.4_{\pm 0.5}$
XLM-R_{large}	$80.4_{\pm 0.6}$	$71.0_{\pm 1.4}$	$82.6_{\pm 0.5}$	$59.4_{\pm 0.8}$

(a) Alignment with bitext used in previous works

	XNLI	NER	POS	Parsing
mBERT	$70.1_{\pm 0.8}$	$67.7_{\pm 1.3}$	$78.3_{\pm 0.5}$	$52.6_{\pm 0.4}$
+ Linear Mapping	$70.2_{\pm 0.6}$	$63.8_{\pm 1.3}$	$80.1_{\pm 0.4}$	$53.6_{\pm 0.3}$
+ L2 Align	$70.3_{\pm 0.5}$	$67.8_{\pm 1.4}$	$78.2_{\pm 1.2}$	$52.8_{\pm 0.7}$
+ Weak Align (Our)	$70.8_{\pm 0.7}$	$67.3_{\pm 0.9}$	$78.8_{\pm 0.6}$	$52.9_{\pm 0.6}$
+ Strong Align (Our)	$70.4_{\pm 0.7}$	$67.2_{\pm 1.1}$	$79.0_{\pm 0.7}$	$53.3_{\pm 0.6}$
XLM-R_{base}	$76.4_{\pm 0.5}$	$66.4_{\pm 0.9}$	$81.2_{\pm 0.6}$	$57.3_{\pm 0.6}$
+ Linear Mapping	$73.5_{\pm 0.5}$	$54.2_{\pm 0.8}$	$81.7_{\pm 0.6}$	$56.1_{\pm 0.4}$
+ L2 Align	$75.8_{\pm 0.5}$	$65.5_{\pm 1.2}$	$81.4_{\pm 0.8}$	$55.9_{\pm 0.6}$
+ Weak Align (Our)	$76.0_{\pm 0.4}$	$66.2_{\pm 1.2}$	$81.5_{\pm 0.5}$	$57.4_{\pm 0.5}$
+ Strong Align (Our)	$76.1_{\pm 0.4}$	$66.2_{\pm 1.0}$	$81.5_{\pm 0.6}$	$57.4_{\pm 0.5}$
XLM-R_{large}	$80.4_{\pm 0.6}$	$71.0_{\pm 1.4}$	$82.6_{\pm 0.5}$	$59.4_{\pm 0.8}$

(b) Alignment with the OPUS-100 bitext

Table 6.2: Zero-shot cross-lingual transfer result, average over 9 languages. Breakdown can be found in Table 6.3 and Table 6.4. Blue or orange indicates the mean performance is one standard derivation above or below the mean of baseline. While mBERT benefits from alignment in some cases, extra alignment does not improve XLM-R.

6.4.3 Findings

6.4.3.1 Robustness of Previous Methods

With a more robust evaluation scheme and 1 million parallel sentences (4× to 100× of previously considered data), the previously proposed Linear Mapping or L2 Alignment does not consistently outperform a no alignment setting more than one standard deviation in all cases (Table 6.2). With mBERT, L2 Alignment performs comparably to no alignment on all 4 tasks (XNLI, NER, POS tagging, and parsing). Compared to no alignment, Linear Mapping performs much worse on NER, performs better on POS tagging and parsing, and performs comparably on XNLI. While previous work observes small improvements on selected languages and tasks, it likely depends on the randomness during evaluation. Based on a more comprehensive evaluation including 4 tasks and multiple seeds, the previously

[4]We take the weighted average of representations in all layers of the encoder.
[5]We pick 5 random seeds before the experiment and use the same seeds for each task and model.

	ar	de	en	es	fr	hi	ru	vi	zh	AVER
XNLI (Accuracy)										
mBERT	64.2$_{\pm0.9}$	70.5$_{\pm0.2}$	82.5$_{\pm0.3}$	74.2$_{\pm1.2}$	73.8$_{\pm0.8}$	59.4$_{\pm0.7}$	68.3$_{\pm0.9}$	69.6$_{\pm0.7}$	68.6$_{\pm0.9}$	70.1$_{\pm0.8}$
+ Linear Mapping	63.8$_{\pm0.6}$	70.4$_{\pm0.4}$	81.0$_{\pm0.5}$	73.9$_{\pm0.9}$	72.5$_{\pm0.8}$	61.2$_{\pm0.7}$	67.1$_{\pm0.4}$	70.2$_{\pm0.5}$	70.1$_{\pm0.8}$	70.0$_{\pm0.6}$
+ L2 Align	64.1$_{\pm0.4}$	70.0$_{\pm0.7}$	82.2$_{\pm0.4}$	73.9$_{\pm0.5}$	73.8$_{\pm0.2}$	58.5$_{\pm0.3}$	67.9$_{\pm0.4}$	69.4$_{\pm0.6}$	67.9$_{\pm0.4}$	69.7$_{\pm0.4}$
+ Weak Align (Our)	64.9$_{\pm0.8}$	71.0$_{\pm0.8}$	82.3$_{\pm0.4}$	74.6$_{\pm0.7}$	73.8$_{\pm0.4}$	59.8$_{\pm0.3}$	68.5$_{\pm1.0}$	70.3$_{\pm0.8}$	69.4$_{\pm1.0}$	70.5$_{\pm0.7}$
+ Strong Align (Our)	64.8$_{\pm0.8}$	70.5$_{\pm0.9}$	82.3$_{\pm0.5}$	74.4$_{\pm0.6}$	74.1$_{\pm0.7}$	59.8$_{\pm0.9}$	68.2$_{\pm0.6}$	70.1$_{\pm0.8}$	69.0$_{\pm1.0}$	70.4$_{\pm0.7}$
XLM-R$_{\text{base}}$	71.8$_{\pm0.2}$	77.3$_{\pm0.5}$	85.1$_{\pm0.3}$	79.3$_{\pm0.5}$	78.8$_{\pm0.4}$	70.3$_{\pm0.6}$	75.9$_{\pm0.5}$	74.8$_{\pm0.4}$	74.1$_{\pm0.5}$	76.4$_{\pm0.5}$
+ Linear Mapping	69.7$_{\pm0.6}$	74.3$_{\pm0.3}$	82.5$_{\pm0.6}$	76.4$_{\pm0.5}$	75.5$_{\pm0.4}$	67.2$_{\pm0.9}$	73.2$_{\pm0.3}$	72.5$_{\pm0.5}$	68.9$_{\pm1.2}$	73.4$_{\pm0.6}$
+ L2 Align	71.6$_{\pm0.8}$	76.0$_{\pm0.5}$	84.5$_{\pm0.5}$	78.6$_{\pm0.3}$	77.9$_{\pm0.3}$	69.8$_{\pm0.7}$	75.3$_{\pm0.3}$	74.0$_{\pm0.4}$	73.7$_{\pm0.7}$	75.7$_{\pm0.5}$
+ Weak Align (Our)	71.7$_{\pm0.7}$	76.5$_{\pm0.6}$	84.7$_{\pm0.6}$	78.7$_{\pm0.6}$	78.1$_{\pm0.7}$	70.4$_{\pm0.9}$	75.8$_{\pm0.6}$	74.5$_{\pm0.5}$	74.2$_{\pm0.7}$	76.1$_{\pm0.7}$
+ Strong Align (Our)	71.6$_{\pm0.5}$	76.6$_{\pm0.4}$	84.7$_{\pm0.5}$	79.0$_{\pm0.4}$	78.3$_{\pm0.3}$	70.0$_{\pm1.0}$	75.7$_{\pm0.7}$	74.7$_{\pm0.4}$	73.7$_{\pm0.8}$	76.0$_{\pm0.6}$
XLM-R$_{\text{large}}$	77.5$_{\pm0.6}$	81.7$_{\pm0.4}$	88.0$_{\pm0.3}$	83.3$_{\pm0.6}$	82.0$_{\pm0.5}$	75.1$_{\pm0.8}$	79.2$_{\pm0.7}$	78.4$_{\pm0.6}$	78.3$_{\pm0.6}$	80.4$_{\pm0.6}$
NER (Entity-level F1)										
mBERT	42.0$_{\pm2.9}$	79.0$_{\pm0.3}$	84.1$_{\pm0.2}$	73.3$_{\pm2.5}$	78.9$_{\pm0.3}$	65.7$_{\pm1.4}$	65.2$_{\pm1.4}$	69.7$_{\pm1.8}$	51.7$_{\pm0.8}$	67.7$_{\pm1.3}$
+ Linear Mapping	36.9$_{\pm1.1}$	76.1$_{\pm0.4}$	82.8$_{\pm0.1}$	70.4$_{\pm2.1}$	77.4$_{\pm0.7}$	64.5$_{\pm1.4}$	59.5$_{\pm2.5}$	65.2$_{\pm2.7}$	40.5$_{\pm2.0}$	63.7$_{\pm1.5}$
+ L2 Align	39.7$_{\pm1.6}$	77.7$_{\pm0.8}$	84.0$_{\pm0.1}$	72.5$_{\pm1.5}$	79.1$_{\pm0.3}$	63.3$_{\pm1.8}$	64.3$_{\pm1.0}$	71.2$_{\pm0.9}$	52.1$_{\pm1.1}$	67.1$_{\pm1.0}$
+ Weak Align (Our)	42.3$_{\pm2.7}$	78.7$_{\pm0.3}$	84.2$_{\pm0.2}$	71.6$_{\pm2.2}$	79.4$_{\pm0.6}$	67.6$_{\pm1.3}$	64.8$_{\pm0.8}$	70.0$_{\pm2.3}$	52.9$_{\pm0.9}$	68.0$_{\pm1.3}$
+ Strong Align (Our)	40.6$_{\pm1.0}$	78.7$_{\pm0.3}$	84.2$_{\pm0.2}$	72.2$_{\pm2.5}$	79.0$_{\pm0.5}$	67.2$_{\pm0.7}$	64.5$_{\pm1.7}$	70.1$_{\pm2.5}$	52.5$_{\pm0.8}$	67.7$_{\pm1.1}$
XLM-R$_{\text{base}}$	44.0$_{\pm1.3}$	75.0$_{\pm0.3}$	82.2$_{\pm0.2}$	76.0$_{\pm2.4}$	77.6$_{\pm0.7}$	65.7$_{\pm0.6}$	64.1$_{\pm0.7}$	68.0$_{\pm1.2}$	45.1$_{\pm0.8}$	66.4$_{\pm0.9}$
+ Linear Mapping	30.8$_{\pm2.1}$	69.0$_{\pm0.6}$	78.3$_{\pm0.3}$	59.8$_{\pm0.5}$	67.8$_{\pm0.7}$	57.9$_{\pm1.5}$	48.0$_{\pm1.0}$	54.4$_{\pm0.5}$	21.0$_{\pm0.9}$	54.1$_{\pm0.9}$
+ L2 Align	44.9$_{\pm2.1}$	74.9$_{\pm0.6}$	82.1$_{\pm0.3}$	75.0$_{\pm3.1}$	77.1$_{\pm0.6}$	65.5$_{\pm1.3}$	63.2$_{\pm0.3}$	66.3$_{\pm2.2}$	42.4$_{\pm0.7}$	65.7$_{\pm1.2}$
+ Weak Align (Our)	45.6$_{\pm1.4}$	75.0$_{\pm0.5}$	82.2$_{\pm0.2}$	74.2$_{\pm2.4}$	77.2$_{\pm0.8}$	65.8$_{\pm1.1}$	63.6$_{\pm1.1}$	67.6$_{\pm0.7}$	42.8$_{\pm0.6}$	66.0$_{\pm1.0}$
+ Strong Align (Our)	45.7$_{\pm1.7}$	75.1$_{\pm0.6}$	82.1$_{\pm0.3}$	73.5$_{\pm1.7}$	77.2$_{\pm0.6}$	65.8$_{\pm1.7}$	63.7$_{\pm0.5}$	68.1$_{\pm0.8}$	43.2$_{\pm0.4}$	66.1$_{\pm0.9}$
XLM-R$_{\text{large}}$	46.8$_{\pm4.3}$	79.1$_{\pm0.5}$	84.2$_{\pm0.2}$	75.7$_{\pm2.9}$	80.7$_{\pm0.5}$	71.6$_{\pm1.1}$	71.7$_{\pm0.5}$	77.4$_{\pm1.3}$	51.5$_{\pm1.4}$	71.0$_{\pm1.4}$
POS (Accuracy)										
mBERT	60.3$_{\pm0.9}$	90.4$_{\pm0.3}$	96.9$_{\pm0.1}$	87.7$_{\pm0.2}$	88.9$_{\pm0.3}$	68.0$_{\pm0.8}$	82.5$_{\pm0.7}$	62.7$_{\pm0.2}$	67.1$_{\pm1.1}$	78.3$_{\pm0.5}$
+ Linear Mapping	73.6$_{\pm0.7}$	88.2$_{\pm0.5}$	96.3$_{\pm0.0}$	87.4$_{\pm0.1}$	88.9$_{\pm0.3}$	77.3$_{\pm0.6}$	78.0$_{\pm1.0}$	60.4$_{\pm0.5}$	65.7$_{\pm1.3}$	79.5$_{\pm0.5}$
+ L2 Align	63.4$_{\pm2.6}$	89.3$_{\pm0.7}$	96.7$_{\pm0.2}$	86.7$_{\pm0.3}$	87.9$_{\pm0.5}$	65.2$_{\pm3.9}$	83.6$_{\pm0.9}$	62.3$_{\pm0.8}$	66.5$_{\pm1.5}$	78.0$_{\pm1.3}$
+ Weak Align (Our)	61.6$_{\pm2.0}$	90.3$_{\pm0.7}$	96.9$_{\pm0.1}$	87.5$_{\pm0.6}$	88.6$_{\pm0.3}$	70.3$_{\pm0.9}$	83.1$_{\pm0.6}$	63.2$_{\pm0.3}$	68.1$_{\pm0.9}$	78.8$_{\pm0.7}$
+ Strong Align (Our)	61.9$_{\pm2.0}$	90.4$_{\pm0.7}$	96.9$_{\pm0.0}$	87.5$_{\pm0.5}$	88.5$_{\pm0.4}$	71.1$_{\pm1.2}$	83.0$_{\pm0.5}$	63.2$_{\pm0.2}$	68.0$_{\pm0.6}$	79.0$_{\pm0.7}$
XLM-R$_{\text{base}}$	70.2$_{\pm1.6}$	91.6$_{\pm0.3}$	97.5$_{\pm0.0}$	88.5$_{\pm0.2}$	89.4$_{\pm0.3}$	71.7$_{\pm1.3}$	86.1$_{\pm0.3}$	64.5$_{\pm0.5}$	71.4$_{\pm0.5}$	81.2$_{\pm0.6}$
+ Linear Mapping	74.3$_{\pm1.1}$	90.7$_{\pm0.5}$	96.9$_{\pm0.0}$	88.2$_{\pm0.1}$	89.3$_{\pm0.3}$	82.1$_{\pm0.9}$	82.7$_{\pm0.4}$	62.6$_{\pm0.4}$	65.3$_{\pm1.0}$	81.3$_{\pm0.5}$
+ L2 Align	71.1$_{\pm1.8}$	91.4$_{\pm0.3}$	97.4$_{\pm0.0}$	88.2$_{\pm0.2}$	89.0$_{\pm0.3}$	73.0$_{\pm3.8}$	86.6$_{\pm0.2}$	64.4$_{\pm0.4}$	70.8$_{\pm0.8}$	81.3$_{\pm0.9}$
+ Weak Align (Our)	72.8$_{\pm0.7}$	91.1$_{\pm0.2}$	97.4$_{\pm0.0}$	88.3$_{\pm0.2}$	89.2$_{\pm0.2}$	72.4$_{\pm1.6}$	86.4$_{\pm0.1}$	64.7$_{\pm0.4}$	71.6$_{\pm1.2}$	81.5$_{\pm0.5}$
+ Strong Align (Our)	72.5$_{\pm0.9}$	91.1$_{\pm0.3}$	97.4$_{\pm0.0}$	88.3$_{\pm0.2}$	89.1$_{\pm0.1}$	72.0$_{\pm2.1}$	86.4$_{\pm0.1}$	64.8$_{\pm0.4}$	71.4$_{\pm1.1}$	81.4$_{\pm0.6}$
XLM-R$_{\text{large}}$	73.9$_{\pm1.0}$	91.9$_{\pm0.3}$	98.0$_{\pm0.0}$	89.2$_{\pm0.2}$	89.8$_{\pm0.1}$	78.4$_{\pm2.1}$	86.5$_{\pm0.2}$	64.8$_{\pm0.3}$	71.0$_{\pm0.3}$	82.6$_{\pm0.5}$
Parsing (Labeled Attachment Score)										
mBERT	28.8$_{\pm0.4}$	67.8$_{\pm0.5}$	79.7$_{\pm0.1}$	69.1$_{\pm0.1}$	73.3$_{\pm0.2}$	31.0$_{\pm0.5}$	60.2$_{\pm0.6}$	33.5$_{\pm0.5}$	29.5$_{\pm0.4}$	52.6$_{\pm0.4}$
+ Linear Mapping	44.1$_{\pm0.3}$	64.4$_{\pm0.4}$	80.5$_{\pm0.2}$	70.2$_{\pm0.3}$	73.9$_{\pm0.1}$	32.2$_{\pm0.3}$	56.7$_{\pm0.5}$	32.1$_{\pm0.2}$	28.1$_{\pm0.3}$	53.6$_{\pm0.3}$
+ L2 Align	29.6$_{\pm1.6}$	66.9$_{\pm0.2}$	79.2$_{\pm0.2}$	68.2$_{\pm0.4}$	72.5$_{\pm0.5}$	30.8$_{\pm1.9}$	60.0$_{\pm0.6}$	33.3$_{\pm0.4}$	29.5$_{\pm0.4}$	52.2$_{\pm0.7}$
+ Weak Align (Our)	30.7$_{\pm0.9}$	67.6$_{\pm0.6}$	79.8$_{\pm0.1}$	69.7$_{\pm0.4}$	73.6$_{\pm0.4}$	31.2$_{\pm0.8}$	61.3$_{\pm0.7}$	33.5$_{\pm0.6}$	30.5$_{\pm0.6}$	53.1$_{\pm0.6}$
+ Strong Align (Our)	31.2$_{\pm1.1}$	67.5$_{\pm0.4}$	79.8$_{\pm0.1}$	69.4$_{\pm0.3}$	73.4$_{\pm0.5}$	30.7$_{\pm1.5}$	61.3$_{\pm0.8}$	33.5$_{\pm0.6}$	30.0$_{\pm0.5}$	53.0$_{\pm0.6}$
XLM-R$_{\text{base}}$	43.7$_{\pm1.7}$	69.0$_{\pm0.4}$	80.5$_{\pm0.2}$	71.0$_{\pm0.4}$	73.6$_{\pm0.5}$	41.2$_{\pm0.9}$	66.3$_{\pm0.9}$	36.6$_{\pm0.2}$	34.2$_{\pm0.7}$	57.3$_{\pm0.6}$
+ Linear Mapping	47.2$_{\pm0.6}$	66.7$_{\pm0.3}$	81.4$_{\pm0.1}$	72.6$_{\pm0.2}$	74.4$_{\pm0.4}$	41.4$_{\pm0.7}$	60.8$_{\pm0.6}$	34.3$_{\pm0.3}$	21.5$_{\pm1.1}$	55.6$_{\pm0.5}$
+ L2 Align	41.3$_{\pm1.8}$	68.1$_{\pm0.3}$	79.7$_{\pm0.2}$	70.0$_{\pm0.5}$	73.0$_{\pm0.5}$	40.2$_{\pm1.6}$	63.7$_{\pm0.9}$	36.5$_{\pm0.5}$	32.9$_{\pm0.3}$	56.2$_{\pm0.7}$
+ Weak Align (Our)	44.6$_{\pm1.0}$	68.8$_{\pm0.4}$	80.4$_{\pm0.1}$	71.4$_{\pm0.2}$	73.9$_{\pm0.2}$	41.0$_{\pm0.6}$	65.7$_{\pm0.4}$	36.7$_{\pm0.4}$	33.8$_{\pm0.3}$	57.4$_{\pm0.4}$
+ Strong Align (Our)	44.8$_{\pm0.9}$	68.9$_{\pm0.5}$	80.4$_{\pm0.1}$	71.3$_{\pm0.2}$	73.9$_{\pm0.1}$	40.7$_{\pm0.8}$	66.2$_{\pm0.4}$	36.7$_{\pm0.3}$	34.0$_{\pm0.8}$	57.4$_{\pm0.5}$
XLM-R$_{\text{large}}$	48.2$_{\pm1.5}$	67.8$_{\pm0.6}$	82.6$_{\pm0.3}$	73.9$_{\pm0.4}$	76.4$_{\pm0.4}$	41.8$_{\pm2.5}$	69.6$_{\pm0.4}$	38.9$_{\pm0.6}$	35.4$_{\pm0.5}$	59.4$_{\pm0.8}$

Table 6.3: Zero-shot cross-lingual transfer result with bitext from previous works. Blue or orange indicates the mean performance is one standard derivation above or below the mean of baseline.

	ar	de	en	es	fr	hi	ru	vi	zh	AVER
XNLI (Accuracy)										
mBERT	64.2$_{\pm0.9}$	70.5$_{\pm0.2}$	82.5$_{\pm0.3}$	74.2$_{\pm1.2}$	73.8$_{\pm0.8}$	59.4$_{\pm0.7}$	68.3$_{\pm0.9}$	69.6$_{\pm0.7}$	68.6$_{\pm0.9}$	70.1$_{\pm0.8}$
+ Linear Mapping	64.1$_{\pm0.7}$	70.0$_{\pm0.6}$	81.0$_{\pm0.5}$	74.1$_{\pm0.6}$	72.9$_{\pm0.9}$	61.8$_{\pm0.7}$	67.4$_{\pm0.6}$	70.2$_{\pm0.5}$	70.2$_{\pm0.8}$	70.2$_{\pm0.6}$
+ L2 Align	64.3$_{\pm0.5}$	70.7$_{\pm1.0}$	82.5$_{\pm0.5}$	74.3$_{\pm0.3}$	74.0$_{\pm0.4}$	59.3$_{\pm0.4}$	68.6$_{\pm0.7}$	69.7$_{\pm0.4}$	69.1$_{\pm0.5}$	70.3$_{\pm0.5}$
+ Weak Align (Our)	65.1$_{\pm0.9}$	70.9$_{\pm0.6}$	82.6$_{\pm0.5}$	74.9$_{\pm0.6}$	74.1$_{\pm0.4}$	60.3$_{\pm0.6}$	68.9$_{\pm0.8}$	70.6$_{\pm0.6}$	69.6$_{\pm1.0}$	70.8$_{\pm0.7}$
+ Strong Align (Our)	64.7$_{\pm0.9}$	70.8$_{\pm0.7}$	82.4$_{\pm0.1}$	74.5$_{\pm0.7}$	73.9$_{\pm0.7}$	59.6$_{\pm0.6}$	68.5$_{\pm1.1}$	70.4$_{\pm0.6}$	69.1$_{\pm1.0}$	70.4$_{\pm0.7}$
XLM-R$_{\text{base}}$	71.8$_{\pm0.2}$	77.3$_{\pm0.5}$	85.1$_{\pm0.3}$	79.3$_{\pm0.5}$	78.8$_{\pm0.4}$	70.3$_{\pm0.6}$	75.9$_{\pm0.5}$	74.8$_{\pm0.4}$	74.1$_{\pm0.5}$	76.4$_{\pm0.5}$
+ Linear Mapping	69.9$_{\pm0.4}$	74.3$_{\pm0.3}$	82.5$_{\pm0.6}$	76.4$_{\pm0.5}$	75.5$_{\pm0.6}$	67.2$_{\pm1.0}$	72.7$_{\pm0.2}$	72.7$_{\pm0.5}$	70.1$_{\pm0.8}$	73.5$_{\pm0.5}$
+ L2 Align	71.9$_{\pm0.6}$	76.4$_{\pm0.4}$	84.6$_{\pm0.3}$	78.4$_{\pm0.5}$	77.8$_{\pm0.3}$	69.9$_{\pm0.8}$	75.2$_{\pm0.5}$	74.2$_{\pm0.5}$	73.7$_{\pm0.5}$	75.8$_{\pm0.5}$
+ Weak Align (Our)	71.8$_{\pm0.6}$	76.5$_{\pm0.5}$	84.6$_{\pm0.2}$	79.0$_{\pm0.4}$	78.4$_{\pm0.5}$	70.0$_{\pm0.5}$	75.7$_{\pm0.3}$	74.7$_{\pm0.3}$	73.4$_{\pm0.6}$	76.0$_{\pm0.4}$
+ Strong Align (Our)	72.0$_{\pm0.5}$	76.6$_{\pm0.4}$	84.8$_{\pm0.1}$	79.0$_{\pm0.4}$	78.6$_{\pm0.5}$	70.1$_{\pm0.3}$	75.7$_{\pm0.4}$	74.8$_{\pm0.6}$	73.8$_{\pm0.6}$	76.1$_{\pm0.4}$
XLM-R$_{\text{large}}$	77.5$_{\pm0.6}$	81.7$_{\pm0.4}$	88.0$_{\pm0.3}$	83.3$_{\pm0.6}$	82.0$_{\pm0.5}$	75.1$_{\pm0.8}$	79.2$_{\pm0.7}$	78.4$_{\pm0.6}$	78.3$_{\pm0.6}$	80.4$_{\pm0.6}$
NER (Entity-level F1)										
mBERT	42.0$_{\pm2.9}$	79.0$_{\pm0.3}$	84.1$_{\pm0.2}$	73.3$_{\pm2.5}$	78.9$_{\pm0.3}$	65.7$_{\pm1.4}$	65.2$_{\pm1.4}$	69.7$_{\pm1.8}$	51.7$_{\pm0.8}$	67.7$_{\pm1.3}$
+ Linear Mapping	36.9$_{\pm0.9}$	76.2$_{\pm0.3}$	82.8$_{\pm0.1}$	71.2$_{\pm1.5}$	77.4$_{\pm0.7}$	62.4$_{\pm2.2}$	59.6$_{\pm2.4}$	65.4$_{\pm2.6}$	42.3$_{\pm1.4}$	63.8$_{\pm1.3}$
+ L2 Align	41.3$_{\pm3.2}$	78.2$_{\pm1.0}$	84.1$_{\pm0.1}$	73.4$_{\pm2.4}$	79.7$_{\pm0.8}$	64.9$_{\pm1.5}$	64.9$_{\pm1.6}$	71.8$_{\pm0.9}$	52.4$_{\pm1.3}$	67.8$_{\pm1.4}$
+ Weak Align (Our)	40.3$_{\pm1.1}$	78.7$_{\pm0.3}$	84.0$_{\pm0.1}$	70.7$_{\pm2.1}$	79.0$_{\pm0.4}$	67.2$_{\pm1.2}$	64.9$_{\pm1.2}$	69.1$_{\pm0.8}$	52.0$_{\pm1.1}$	67.3$_{\pm0.9}$
+ Strong Align (Our)	40.7$_{\pm1.9}$	78.3$_{\pm0.3}$	84.2$_{\pm0.1}$	70.0$_{\pm2.6}$	78.8$_{\pm0.3}$	66.7$_{\pm1.4}$	64.8$_{\pm0.9}$	69.5$_{\pm1.4}$	52.1$_{\pm0.6}$	67.2$_{\pm1.1}$
XLM-R$_{\text{base}}$	44.0$_{\pm1.3}$	75.0$_{\pm0.3}$	82.2$_{\pm0.2}$	76.0$_{\pm2.4}$	77.6$_{\pm0.7}$	65.7$_{\pm0.6}$	64.1$_{\pm0.7}$	68.0$_{\pm1.2}$	45.1$_{\pm0.8}$	66.4$_{\pm0.9}$
+ Linear Mapping	30.8$_{\pm1.6}$	69.3$_{\pm0.6}$	78.3$_{\pm0.3}$	60.2$_{\pm0.8}$	67.9$_{\pm0.5}$	58.2$_{\pm0.7}$	47.7$_{\pm0.8}$	54.1$_{\pm0.3}$	21.6$_{\pm1.2}$	54.2$_{\pm0.8}$
+ L2 Align	44.1$_{\pm1.2}$	74.2$_{\pm0.7}$	81.9$_{\pm0.3}$	74.9$_{\pm3.3}$	76.9$_{\pm0.6}$	64.7$_{\pm0.5}$	61.9$_{\pm1.4}$	68.4$_{\pm2.2}$	42.1$_{\pm1.1}$	65.5$_{\pm1.2}$
+ Weak Align (Our)	45.5$_{\pm2.8}$	75.0$_{\pm0.8}$	82.2$_{\pm0.2}$	73.7$_{\pm1.8}$	77.3$_{\pm0.6}$	66.6$_{\pm1.3}$	64.0$_{\pm1.2}$	67.5$_{\pm1.4}$	43.9$_{\pm1.2}$	66.2$_{\pm1.2}$
+ Strong Align (Our)	45.3$_{\pm1.5}$	75.1$_{\pm0.4}$	82.2$_{\pm0.2}$	74.6$_{\pm2.5}$	77.4$_{\pm0.6}$	66.0$_{\pm1.2}$	63.7$_{\pm0.9}$	68.0$_{\pm1.1}$	43.3$_{\pm0.4}$	66.2$_{\pm1.0}$
XLM-R$_{\text{large}}$	46.8$_{\pm4.3}$	79.1$_{\pm0.5}$	84.2$_{\pm0.2}$	75.7$_{\pm2.9}$	80.7$_{\pm0.5}$	71.6$_{\pm1.1}$	71.7$_{\pm0.5}$	77.4$_{\pm1.3}$	51.5$_{\pm1.4}$	71.0$_{\pm1.4}$
POS (Accuracy)										
mBERT	60.3$_{\pm0.9}$	90.4$_{\pm0.3}$	96.9$_{\pm0.1}$	87.7$_{\pm0.2}$	88.9$_{\pm0.3}$	68.0$_{\pm0.8}$	82.5$_{\pm0.7}$	62.7$_{\pm0.2}$	67.1$_{\pm1.1}$	78.3$_{\pm0.5}$
+ Linear Mapping	76.2$_{\pm0.5}$	91.2$_{\pm0.1}$	96.3$_{\pm0.0}$	87.6$_{\pm0.1}$	89.0$_{\pm0.2}$	74.9$_{\pm1.1}$	80.6$_{\pm0.3}$	60.4$_{\pm0.5}$	64.8$_{\pm1.3}$	80.1$_{\pm0.4}$
+ L2 Align	62.7$_{\pm2.9}$	89.5$_{\pm0.8}$	96.8$_{\pm0.1}$	87.1$_{\pm0.3}$	88.3$_{\pm0.2}$	65.2$_{\pm3.7}$	83.8$_{\pm1.0}$	62.8$_{\pm0.5}$	67.3$_{\pm1.1}$	78.2$_{\pm1.2}$
+ Weak Align (Our)	61.1$_{\pm1.3}$	90.4$_{\pm0.8}$	96.9$_{\pm0.0}$	87.7$_{\pm0.5}$	88.7$_{\pm0.3}$	70.3$_{\pm1.2}$	83.2$_{\pm0.6}$	63.3$_{\pm0.3}$	68.0$_{\pm0.5}$	78.8$_{\pm0.6}$
+ Strong Align (Our)	61.7$_{\pm1.7}$	90.5$_{\pm0.7}$	96.9$_{\pm0.0}$	87.7$_{\pm0.6}$	88.7$_{\pm0.4}$	70.5$_{\pm1.0}$	83.3$_{\pm0.7}$	63.1$_{\pm0.3}$	68.2$_{\pm0.8}$	79.0$_{\pm0.7}$
XLM-R$_{\text{base}}$	70.2$_{\pm1.6}$	91.6$_{\pm0.3}$	97.5$_{\pm0.0}$	88.5$_{\pm0.2}$	89.4$_{\pm0.3}$	71.7$_{\pm1.3}$	86.1$_{\pm0.3}$	64.5$_{\pm0.5}$	71.4$_{\pm0.5}$	81.2$_{\pm0.6}$
+ Linear Mapping	76.0$_{\pm0.9}$	92.0$_{\pm0.1}$	96.9$_{\pm0.0}$	88.7$_{\pm0.2}$	89.5$_{\pm0.3}$	78.9$_{\pm2.1}$	83.9$_{\pm0.3}$	62.5$_{\pm0.4}$	66.5$_{\pm1.0}$	81.7$_{\pm0.6}$
+ L2 Align	71.0$_{\pm0.9}$	91.2$_{\pm0.5}$	97.3$_{\pm0.0}$	87.9$_{\pm0.3}$	88.8$_{\pm0.4}$	74.8$_{\pm2.9}$	86.9$_{\pm0.8}$	64.0$_{\pm0.6}$	70.6$_{\pm0.5}$	81.4$_{\pm0.8}$
+ Weak Align (Our)	72.5$_{\pm0.8}$	91.2$_{\pm0.3}$	97.4$_{\pm0.0}$	88.2$_{\pm0.2}$	89.2$_{\pm0.2}$	72.7$_{\pm1.3}$	86.2$_{\pm0.2}$	64.7$_{\pm0.4}$	71.8$_{\pm1.4}$	81.5$_{\pm0.5}$
+ Strong Align (Our)	72.5$_{\pm0.6}$	91.2$_{\pm0.2}$	97.4$_{\pm0.1}$	88.3$_{\pm0.2}$	89.2$_{\pm0.2}$	72.0$_{\pm1.9}$	86.5$_{\pm0.2}$	64.8$_{\pm0.4}$	71.7$_{\pm1.7}$	81.5$_{\pm0.6}$
XLM-R$_{\text{large}}$	73.9$_{\pm1.0}$	91.9$_{\pm0.3}$	98.0$_{\pm0.0}$	89.2$_{\pm0.2}$	89.8$_{\pm0.1}$	78.4$_{\pm2.1}$	86.5$_{\pm0.2}$	64.8$_{\pm0.3}$	71.0$_{\pm0.3}$	82.6$_{\pm0.5}$
Parsing (Labeled Attachment Score)										
mBERT	28.8$_{\pm0.4}$	67.8$_{\pm0.5}$	79.7$_{\pm0.1}$	69.1$_{\pm0.1}$	73.3$_{\pm0.2}$	31.0$_{\pm0.5}$	60.2$_{\pm0.6}$	33.5$_{\pm0.5}$	29.5$_{\pm0.4}$	52.6$_{\pm0.4}$
+ Linear Mapping	45.0$_{\pm0.3}$	67.7$_{\pm0.2}$	80.5$_{\pm0.2}$	70.0$_{\pm0.3}$	73.9$_{\pm0.2}$	28.4$_{\pm0.2}$	57.2$_{\pm0.4}$	32.0$_{\pm0.3}$	28.1$_{\pm0.2}$	53.6$_{\pm0.3}$
+ L2 Align	29.7$_{\pm0.6}$	67.7$_{\pm0.7}$	79.3$_{\pm0.4}$	68.9$_{\pm0.6}$	73.4$_{\pm0.5}$	31.7$_{\pm1.8}$	61.3$_{\pm1.2}$	33.6$_{\pm0.5}$	29.7$_{\pm0.2}$	52.8$_{\pm0.7}$
+ Weak Align (Our)	29.9$_{\pm1.0}$	67.6$_{\pm0.4}$	79.8$_{\pm0.0}$	69.6$_{\pm0.3}$	73.5$_{\pm0.5}$	31.0$_{\pm1.6}$	61.2$_{\pm0.9}$	33.4$_{\pm0.7}$	30.0$_{\pm0.5}$	52.9$_{\pm0.6}$
+ Strong Align (Our)	30.8$_{\pm0.9}$	68.0$_{\pm0.4}$	79.8$_{\pm0.1}$	69.9$_{\pm0.3}$	73.7$_{\pm0.5}$	31.5$_{\pm1.5}$	61.8$_{\pm0.6}$	33.5$_{\pm0.6}$	30.4$_{\pm0.4}$	53.3$_{\pm0.6}$
XLM-R$_{\text{base}}$	43.7$_{\pm1.7}$	69.0$_{\pm0.4}$	80.5$_{\pm0.2}$	71.0$_{\pm0.4}$	73.6$_{\pm0.5}$	41.2$_{\pm0.9}$	66.3$_{\pm0.9}$	36.6$_{\pm0.2}$	34.2$_{\pm0.7}$	57.3$_{\pm0.6}$
+ Linear Mapping	48.0$_{\pm0.5}$	69.2$_{\pm0.2}$	81.4$_{\pm0.1}$	72.4$_{\pm0.1}$	74.8$_{\pm0.3}$	38.8$_{\pm0.9}$	61.8$_{\pm0.5}$	34.2$_{\pm0.3}$	24.2$_{\pm0.9}$	56.1$_{\pm0.4}$
+ L2 Align	39.4$_{\pm0.5}$	68.0$_{\pm0.5}$	79.9$_{\pm0.2}$	69.9$_{\pm0.5}$	72.8$_{\pm0.5}$	40.2$_{\pm1.1}$	63.8$_{\pm0.8}$	36.4$_{\pm0.6}$	32.3$_{\pm0.9}$	55.9$_{\pm0.6}$
+ Weak Align (Our)	44.5$_{\pm1.3}$	68.7$_{\pm0.7}$	80.4$_{\pm0.1}$	71.3$_{\pm0.3}$	73.8$_{\pm0.3}$	41.4$_{\pm0.8}$	65.7$_{\pm0.4}$	36.7$_{\pm0.4}$	34.0$_{\pm0.7}$	57.4$_{\pm0.5}$
+ Strong Align (Our)	44.9$_{\pm1.0}$	68.8$_{\pm0.6}$	80.4$_{\pm0.1}$	71.2$_{\pm0.2}$	73.8$_{\pm0.2}$	41.1$_{\pm0.8}$	65.9$_{\pm0.5}$	36.6$_{\pm0.3}$	33.9$_{\pm0.7}$	57.4$_{\pm0.5}$
XLM-R$_{\text{large}}$	48.2$_{\pm1.5}$	67.8$_{\pm0.6}$	82.6$_{\pm0.3}$	73.9$_{\pm0.4}$	76.4$_{\pm0.4}$	41.8$_{\pm2.5}$	69.6$_{\pm0.4}$	38.9$_{\pm0.6}$	35.4$_{\pm0.5}$	59.4$_{\pm0.8}$

Table 6.4: Zero-shot cross-lingual transfer result with the OPUS-100 bitext. Blue or orange indicates the mean performance is one standard derivation above or below the mean of baseline.

proposed methods do not consistently perform better than no alignment with millions of parallel sentences.

6.4.3.2 Contrastive Alignment

In Table 6.2, with mBERT, both proposed contrastive alignment methods consistently perform as well as no alignment while outperforming more than 1 standard deviation on POS tagging and/or parsing. This suggests the proposed methods are more robust to suboptimal alignments. We hypothesize that learned cosine similarity and contrastive alignment allow the model to recover from suboptimal alignments. Both weak and strong alignment perform comparably.

6.4.3.3 Alignment with XLM-R

XLM-R, trained on 2.5TB of text, has the same number of transformer layers as mBERT but with a larger vocabulary. It performs much better than mBERT. Therefore, we wonder if an explicit alignment objective can similarly lead to better cross-lingual representations. Unfortunately, in Table 6.2, we find all alignment methods we consider do not improve over no alignment. Compared to no alignment, Linear Mapping and L2 Alignment have worse performance in 3 out of 4 tasks (except POS tagging). In contrast to previous work, both contrastive alignment objectives perform comparably to no alignment in all 4 tasks.

6.4.3.4 Impact of Bitext Quality

Even though the OPUS-100 bitext has lower quality compared to bitext used in previous works (due to its greater inclusion of bitext from various sources), by comparing Table 6.2a and Table 6.2b, we observe that it has minimum impact on each alignment method we consider. This is good news for the lower resource languages, as not all languages are covered by MultiUN or Europarl.

6.4.3.5 Model Capacity vs Alignment

XLM-R_{large} has nearly twice the number of parameters as XLM-R_{base}. Even trained on the same data, it performs much better than XLM-R_{base}, with or without alignment, as shown in Table 6.2. This suggests increasing model capacity likely leads to better cross-lingual representations than using an explicit alignment objective.

6.5 Discussion

In this chapter, we discuss how to inject cross-lingual signals into multilingual encoders. For type-level cross-lingual signal like bilingual dictionary, we show that adding additional subwords overlap by creating synthetic code-switching corpus with bilingual dictionary benefits cross-lingual representation. For sentence-level cross-lingual signal like bitext, we propose contrastive alignment objective and show that it outperforms L2 Alignment (Cao, Kitaev, and Klein, 2020) and consistently performs as well as or better than no alignment

using various quality bitext on 4 NLP tasks under a comprehensive evaluation with multiple seeds.

However, to our surprise, previously proposed methods do not show consistent improvement over no alignment in this setting. Therefore, we make the following recommendations for future work on cross-lingual alignment or multilingual representations: 1) Evaluations should consider average quality data, not exclusively high-quality bitext. 2) Evaluation must consider multiple NLP tasks or datasets. 3) Evaluation should report **mean and variance over multiple seeds**, not a single run. More broadly, the community must establish a robust evaluation scheme for zero-shot cross-lingual transfer as a single run with one random seed does not reflect the variance of the method (especially in a zero-shot or few-shot setting).[6] While Keung et al. (2020) advocate using oracle for model selection, we instead argue reporting the variance of test performance, following the few-shot learning literature.

Finally, no explicit alignment methods with bitext improve XLM-R and the larger XLM-R_{large} performs much better. While bilingual dictionary contributes to improved cross-lingual representation for 8-layers encoders, the performance gain is likely eclipsed by scaling up the model size. Indeed, as Kale et al. (2021) find that the gain from incorporating bitext into pretraining decreases as model size increase. For smaller model, incorporating cross-lingual signal explicitly might still offers good performance gain. However, as raw text is easier to obtain than bitext, scaling models to more raw text and larger capacity models may be more beneficial for producing better cross-lingual models, as evidenced by Xue et al. (2021) and

[6]This includes zero-shot cross-lingual transfer benchmarks like XGLUE (Liang et al., 2020) and XTREME (Hu et al., 2020).

Goyal et al. (2021).

In this chapter, we observe that zero-shot cross-lingual transfer has low variance on source language generalization performance but high variance on target language generalization performance (Table 6.2a or Table 6.2b). In chapter 7, we will investigate why zero-shot cross-lingual transfer has high variance.

Chapter 7

Why Does Zero-shot Cross-lingual Transfer Have High Variance?

7.1 Introduction

In chapter 6, we observe that While the generalization performance on the source language has low variance, on the target language the variance is much higher with zero-shot cross-lingual transfer, making it difficult to compare different models in the literature and meta-benchmark. Similarly, pretrained monolingual encoders also have unstable performance during fine-tuning (Devlin et al., 2019; Phang, Févry, and Bowman, 2018).

Why are these models so sensitive to the random seed? Many theories have bee offered: catastrophic forgetting of the pretrained task (Phang, Févry, and Bowman, 2018; Lee, Cho, and Kang, 2020; Keung et al., 2020), small data size (Devlin et al., 2019), impact of random seed on task-specific layer initialization and data ordering (Dodge et al., 2020), the Adam optimizer without bias correction (Mosbach, Andriushchenko, and Klakow, 2021; Zhang et al., 2021), and a different generalization error with similar training loss (Mosbach, Andriushchenko, and Klakow, 2021). However, none of these factors fully explain the high generalization error variance of zero-shot cross-lingual transfer on target language but low variance on source language.

In this chapter, we offer a new explanation for high variance in target language performance: *the zero-shot cross-lingual transfer optimization problem is under-specified*. Based on the well-established linear interpolation of 1-dimensional plot and contour plot (Goodfellow, Vinyals, and Saxe, 2014; Li et al., 2018), we empirically show that any linear-interpolated model between the monolingual source model and bilingual source and target model has equally low source language generation error. Yet the target language generation

error surprisingly reduces smoothly and linearly as we move from a monolingual model to a bilingual model. To the best of our knowledge, no other paper documents this finding.

This result provides a new answer to our mystery: only a small subset of the solution space for the source language solves the target language on par with models with actual target language supervision; the optimization could not find such a solution without target language supervision, hence an under-specified optimization problem. If target language supervision were available, as it was in the counterfactual bilingual model, the optimization finds the smaller subset. By comparing both mBERT and XLM-R, we find that the generalization error surface of XLM-R is flatter than mBERT, contributing to its better performance compared to mBERT. Thus, zero-shot cross-lingual transfer has high variance, as the solution found by zero-shot cross-lingual transfer lies in the non-flat region of the target language generalization error surface.

7.2 Existing Hypotheses

Prior studies have observed encoder model instability, and have offered various hypotheses to explain this behavior. Catastrophic forgetting – when neural networks trained on one task forget that task after training on a second task (McCloskey and Cohen, 1989; Kirkpatrick et al., 2017) —has been credited as the source of high variance in both monolingual fine-tuning (Phang, Févry, and Bowman, 2018; Lee, Cho, and Kang, 2020) and zero-shot cross-lingual transfer (Keung et al., 2020). Mosbach, Andriushchenko, and Klakow (2021)

wonder why preserving cloze capability is important. In zero-shot cross-lingual transfer, deliberately preserving the multilingual cloze capability with regularization improves performance but does not eliminate the zero-shot transfer gap (Aghajanyan et al., 2021; Liu et al., 2021b).

Small training data size often seems to have higher variance in performance (Devlin et al., 2019), but Mosbach, Andriushchenko, and Klakow (2021) found that when controlling the number of gradient updates, smaller data size has the similar variance as larger data size.

In the pretraining-then-fine-tune paradigm, random seeds mainly impact the initialization of task-specific layers and data ordering during fine-tuning. Dodge et al. (2020) show development set performance has high variance with respect to seeds. Additionally, Adam optimizer without bias correction—an Adam (Kingma and Ba, 2014) variant (inadvertently) introduced by the implementation of Devlin et al. (2019)—has been identified as the source of high variance during monolingual fine-tuning (Mosbach, Andriushchenko, and Klakow, 2021; Zhang et al., 2021). However, in zero-shot cross-lingual transfer, while different random seeds lead to high variance in target languages, the source language has much smaller variance in comparison even with standard Adam (Wu and Dredze, 2020b).

Beyond optimizers, Mosbach, Andriushchenko, and Klakow (2021) attributes high variance to generalization issues: despite having similar training loss, different models exhibit vastly different development set performance. However, in zero-shot cross-lingual transfer, the development or test performance variance is much smaller on the source language compared to target language.

7.3 Zero-shot Cross-lingual Transfer is Under-specified Optimization

Existing hypotheses do not explain the high variance of zero-shot cross-lingual transfer: much higher variance on generalization error of the target language compared to the source language. We propose a new explanation: *zero-shot cross-lingual transfer is an under-specified optimization problem*. Optimizing a multilingual model for a specific task using only source language annotation allows choices of many good solutions in terms of generalization error. However, unbeknownst to the optimizer, these solutions have wildly different generalization errors performance on the target language. In fact, a small subset has similar low generalization error as models trained on target language. Yet without the guidance of target data, the zero-shot cross-lingual optimization could not find this smaller subset. As we will show in §7.5, the solution found by zero-shot transfer lies in a non-flat region of target language generalization error, causing its high variance.

7.3.1 Linear Interpolation

We test this hypothesis via a linear interpolation between two models to explore the neural network parameter space. Consider three sets of neural network parameters: θ_{src}, θ_{tgt}, $\theta_{\{src,tgt\}}$ for a model trained on task data for the source language only, target language

only and both languages, respectively. This includes both task-specific layers and encoders.[1] Note all three models have the same initialization before fine-tuning, making the bilingual model a counterfactual setup if the corresponding target language supervision was available. We obtain the 1-dimensional (1D) linear interpolation of a monolingual (source) task trained model and bilingual task trained model with

$$\theta(\alpha) = \alpha\theta_{\{src,tgt\}} + (1-\alpha)\theta_{src} \tag{7.1}$$

or we could swap source and target by

$$\theta(\alpha) = \alpha\theta_{\{src,tgt\}} + (1-\alpha)\theta_{tgt} \tag{7.2}$$

where α is a scalar mixing coefficient (Goodfellow, Vinyals, and Saxe, 2014). Additionally, we can compute a 2-dimensional linear interpolation as

$$\theta(\alpha_1, \alpha_2) = \theta_{\{src,tgt\}} + \alpha_1\delta_{src} + \alpha_2\delta_{tgt} \tag{7.3}$$

where $\delta_{src} = \theta_{src} - \theta_{\{src,tgt\}}$, $\delta_{tgt} = \theta_{tgt} - \theta_{\{src,tgt\}}$, α_1 and α_2 are scalar mixing coefficients (Li et al., 2018).[2] Finally, we can evaluate any interpolated models on the

[1] We experiment with interpolating the encoder parameters only and observe similar findings. On the other hand, interpolating the task-specific layer only has a negligible effect.

[2] Li et al. (2018) use two random directions and they normalize it to compensate scaling issue. In this setup, we find δ_{src} and δ_{tgt} have near identical norms, so we do not apply additional normalization. As these two directions are not random, we find that it spans around 55°. We plot the norm ratio and angle of these two vectors in Figure 7.1.

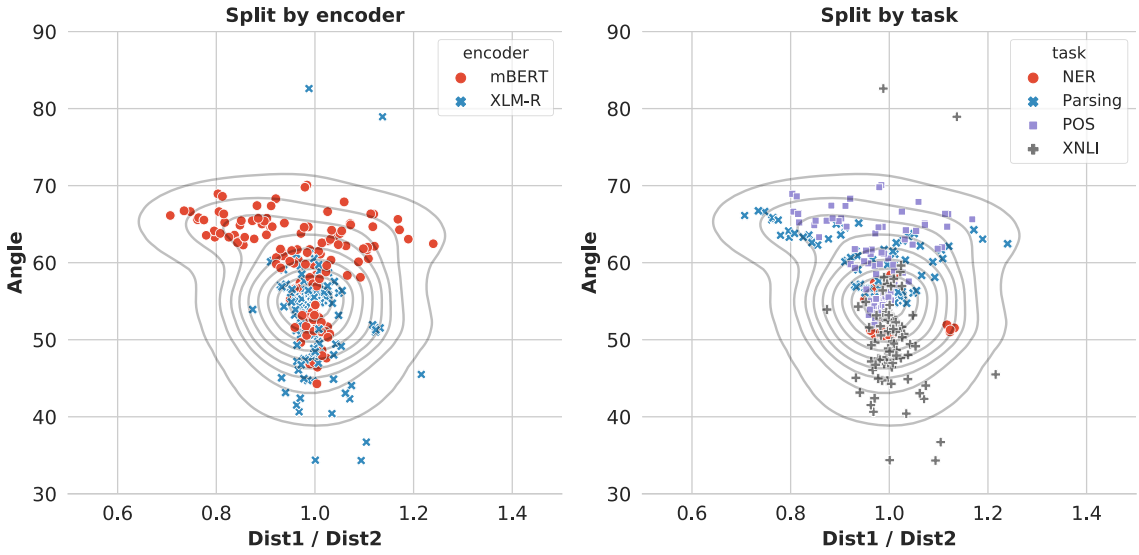


Figure 7.1: $\|\delta_{src}\|/\|\delta_{tgt}\|$ v.s. angle between δ_{src} and δ_{tgt}. Most δ_{src} and δ_{tgt} have similar norms, and the angle between them is around 55°.

development set of source and target languages, testing the generalization error on the same language and across languages.

The performance of the interpolated model illuminates the behavior of the model's parameters. Take Equation 7.1 as an example: if the linear interpolated model performs consistently high for our task on the source language, it suggests that both models lie within the same local minimum of source language generalization error surface. Additionally, if the linear interpolated model performs vastly differently on the target language, it would support our hypothesis. On the other hand, if the linear interpolated model performance drops on the source language, it suggests that both models lie in different local minimum of source language generalization error surface, suggesting the zero-shot optimization searching the wrong region.

7.4 Experiments

We consider four tasks: natural language inference (XNLI; Conneau et al., 2018), named entity recognition (NER; Pan et al., 2017), POS tagging and dependency parsing (Zeman, 2020b). We evaluate XNLI and POS tagging with accuracy (ACC), NER with span-level F1, and parsing with labeled attachment score (LAS). We consider two encoders: base mBERT and large XLM-R. For the task-specific layer, we use a linear classifier for XNLI, NER, and POS tagging, and Dozat and Manning (2017) for dependency parsing.

To avoid English-centric experiments, we consider two source languages: English and Arabic. We choose 8 topologically diverse target languages: Arabic[3], German, Spanish, French, Hindi, Russian, Vietnamese, and Chinese. We train the source language only and target language only monolingual model as well as a source-target bilingual model.

We compute the linear interpolated models as described in Section 7.3.1 and test it on both the source and target language development set. We loop over $\{-0.5, -0.4, \cdots, 1.5\}$ for α, α_1 and α_2.[4] We report the mean and variance of three runs by using different random seeds. We normalized both mean and variance of each interpolated model by the bilingual model performance, allowing us to aggregate across tasks and language pairs.

We follow the implementation and hyperparameter of chapter 6. We optimize with Adam (Kingma and Ba, 2014). The learning rate is `2e-5`. The learning rate scheduler has 10% steps linear warmup then linear decay till 0. We train for 5 epochs and the batch size is

[3]Arabic is only used when English is the source language.

[4]We additionally select 0.025, 0.05, 0.075, 0.125, 0.15, 0.175, 0.825, 0.85, 0.875, 0.925, 0.95, and 0.975 for α due to preliminary experiment.

32. For token level tasks, the task-specific layer takes the representation of the first subword, following previous chapters. Model selection is done on the corresponding dev set of the training set.

During fine-tuning, the maximum sequence length is 128. We use a sliding window of context to include subwords beyond the first 128 for NER and POS tagging. At test time, we use the same maximum sequence length with the exception of parsing, where the first 128 words instead of subwords of a sentence were used. We ignore words with POS tags of `SYM` and `PUNCT` during parsing evaluation. For NER, we adapt the same post-processing as Section 3.2.3. For POS tagging and dependency parsing, we use the following treebanks: Arabic-PADT, German-GSD, English-EWT, Spanish-GSD, French-GSD, Hindi-HDTB, Russian-GSD, Vietnamese-VTB, and Chinese-GSD. Since the Chinese NER is labeled on character-level (including code-switched portion), we segment the Chinese character into word using the Stanford Word Segmenter and realign the label.

7.5 Findings

In Figure 7.2, we observe that interpolations between the source monolingual and bilingual model have consistently similar source language performance. In contrast, surprisingly, the target language performance smoothly and linearly improves as the interpolated model moves from the zero-shot model to bilingual model.[5] Break down of Figure 7.2 by task

[5]We also show the variance of the interpolated models in Figure 7.3. The source language has much lower variance compared to target language on the monolingual side of the interpolated models, echoing findings in chapter 6.

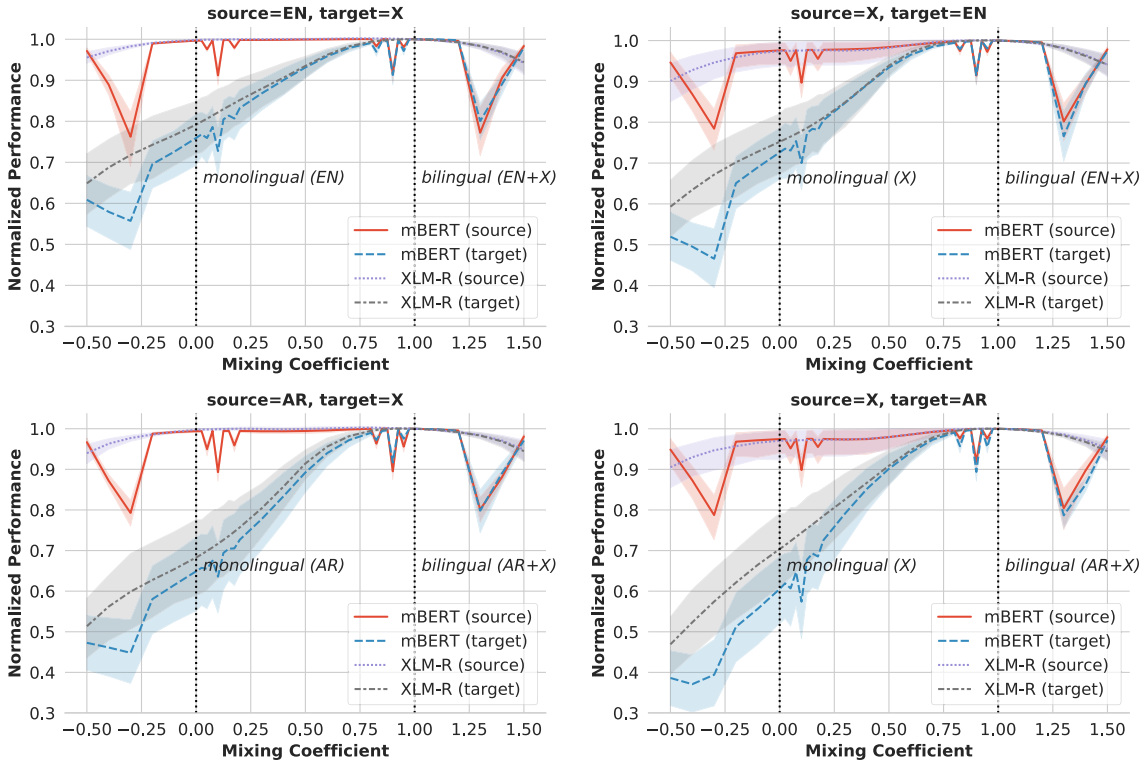


Figure 7.2: Normalized performance of a linear interpolated model between a monolingual and bilingual model. A single plot line shows the performance normalized by the matching bilingual model and aggregated over eight language pairs and four tasks, with the shaded region representing 95% confidence interval. The x-axis is the linear mixing coefficient α in Equation 7.1 and Equation 7.2, with $\alpha = 0$ and $\alpha = 1$ representing source language monolingual model and source + target bilingual model, respectively. To allow aggregating, for each encoder, language pair and task combination, we normalized the interpolated model performance by its corresponding bilingual performance. Each subfigure title indicates the source and target languages. Across all experiments, the source language dev performance stays consistently high (red and purple lines) during interpolation while the target language dev performance starts low and increases smoothly and linearly as it moves towards the bilingual model (gray and blue lines). Break down of this figure by tasks can be found in Figure 7.4a (NER), Figure 7.4b (Parsing), Figure 7.5a (POS), and Figure 7.5b (XNLI), and we observe similar findings.

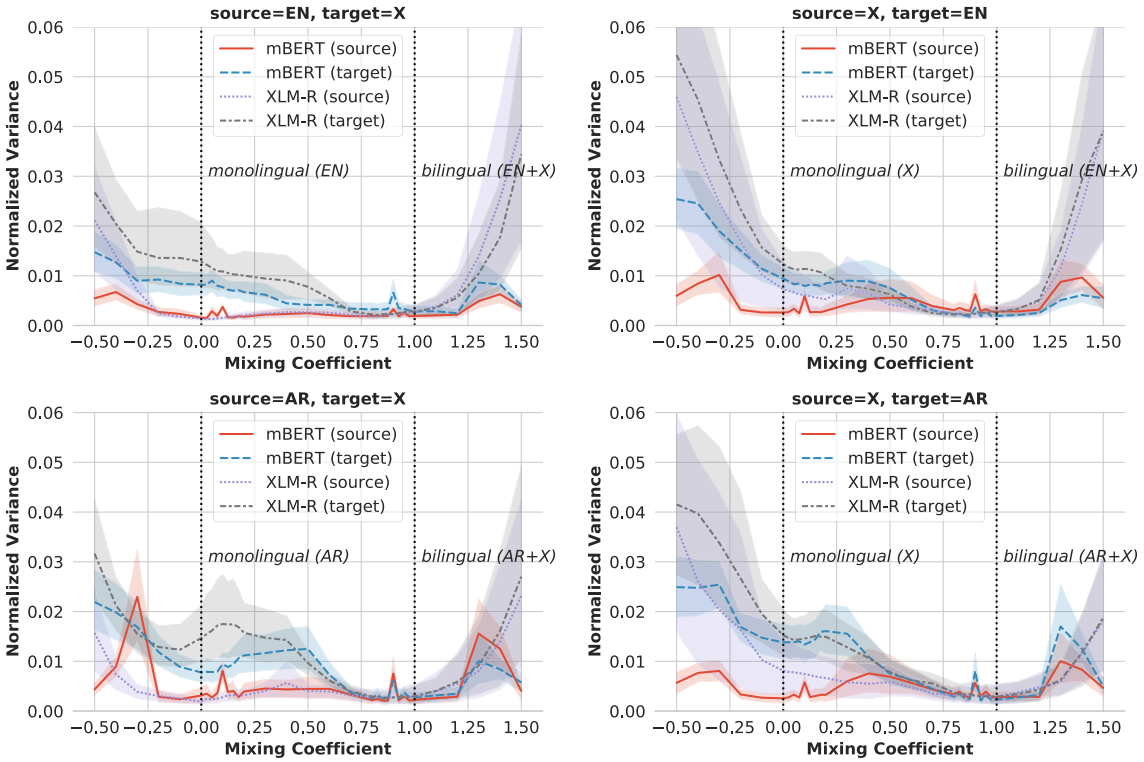
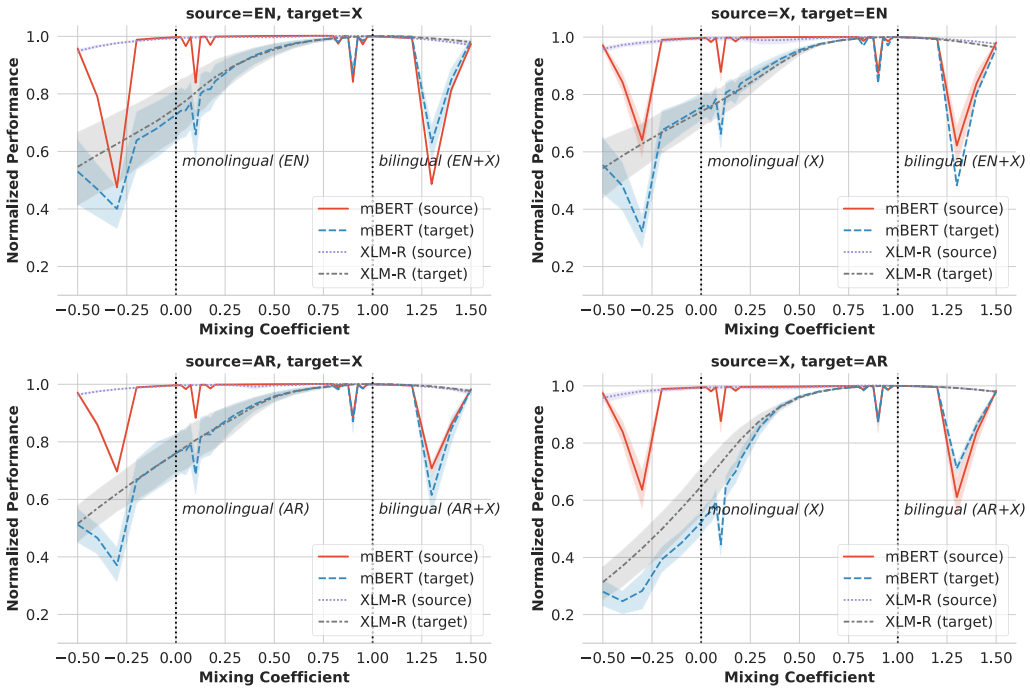
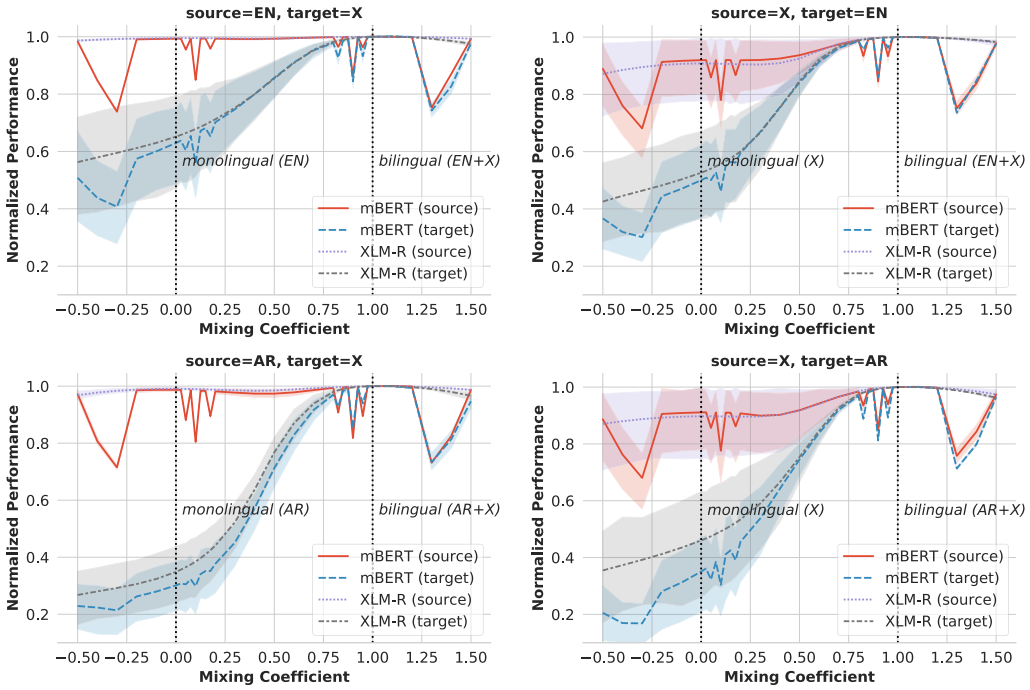


Figure 7.3: Normalized variance of linear interpolation between monolingual model and bilingual model. The source language has much lower variance compared to target language on the monolingual side of the interpolated models.

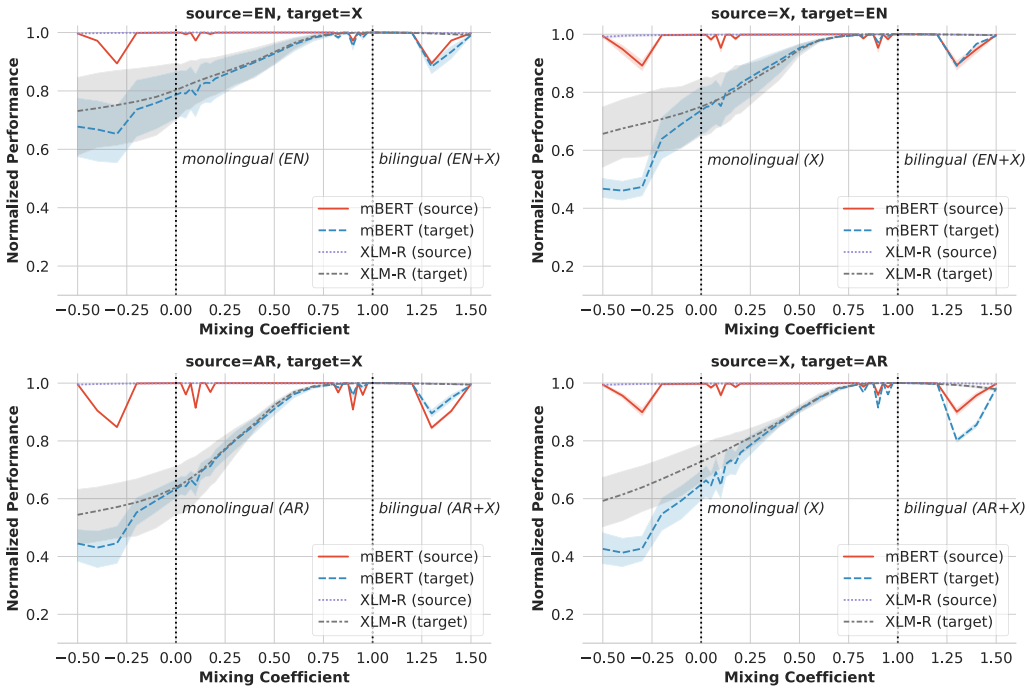


(a) NER

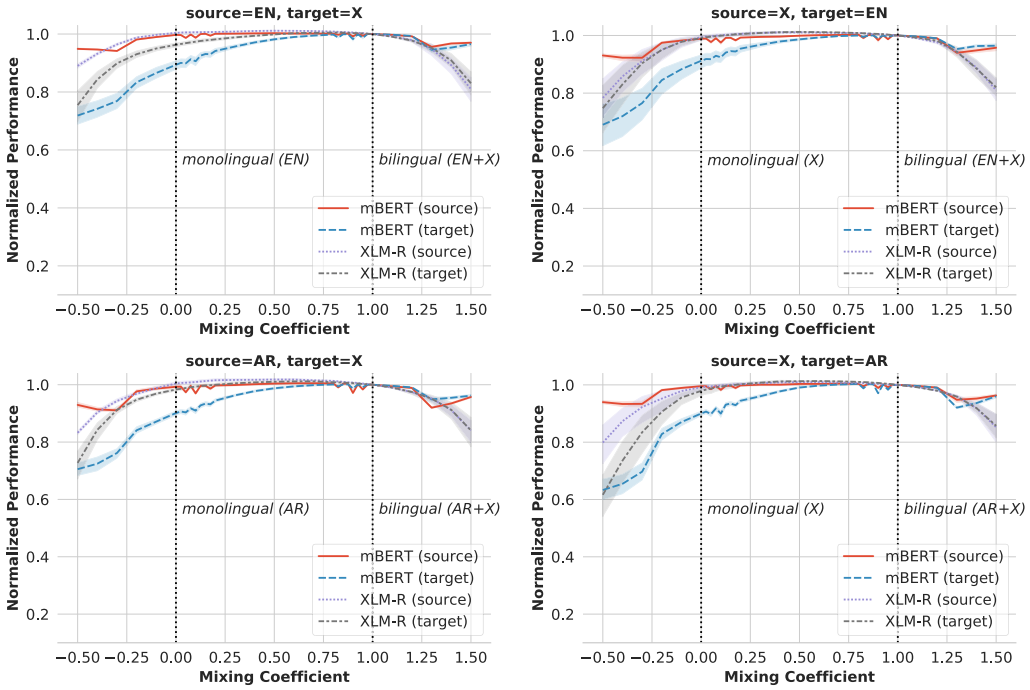


(b) Parsing

Figure 7.4: Normalized NER and Parsing performance of linear interpolated model between monolingual and bilingual model



(a) POS



(b) XNLI

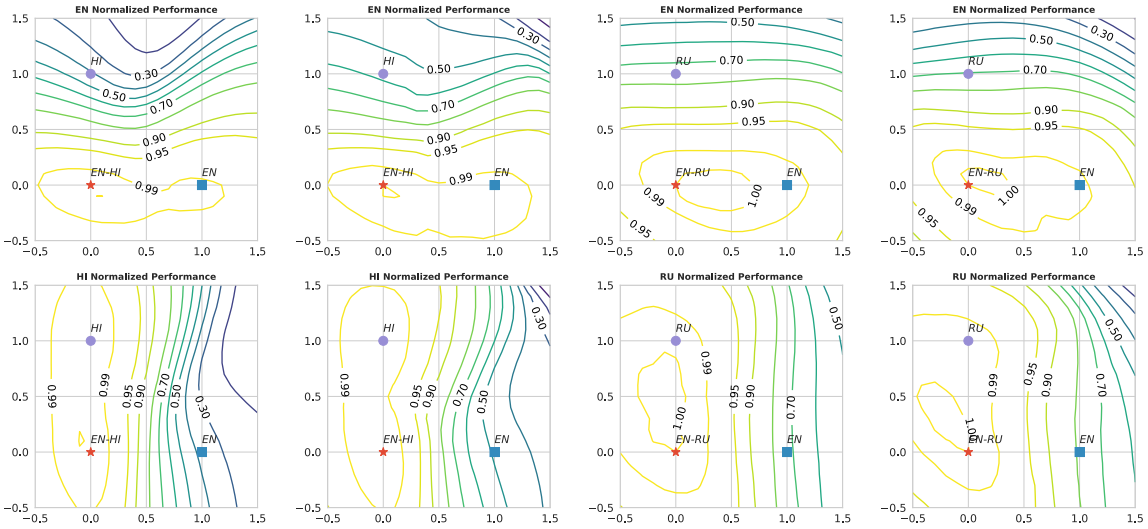
Figure 7.5: Normalized POS and XNLI performance of linear interpolated model between monolingual and bilingual model

shows different tasks follow similar trends. The only exception is mBERT, where the performance drops slightly around 0.1 and 0.9 locally. In contrast, XLM-R has a flatter slope and smoother interpolated models.

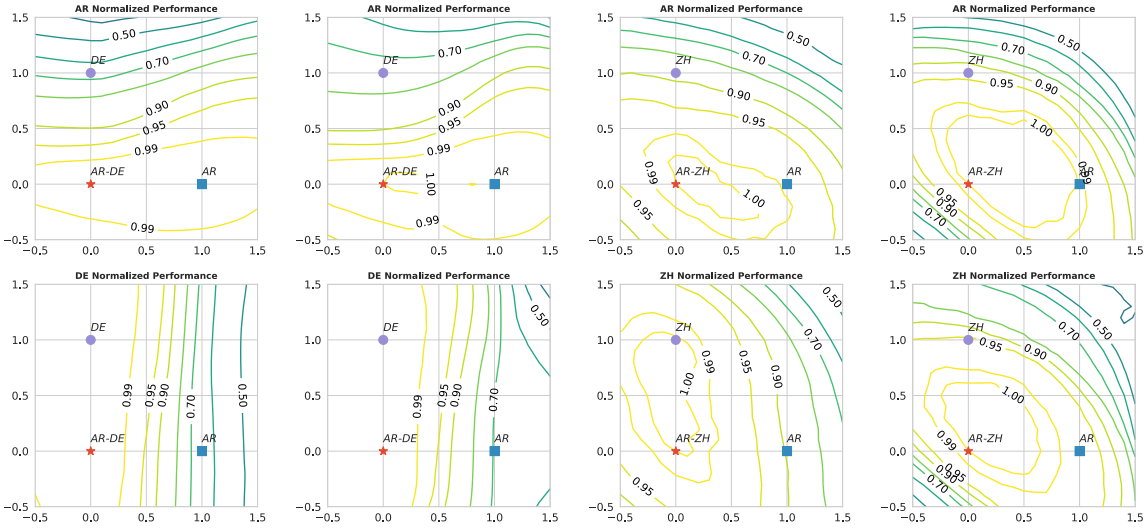
Figure 7.6 further demonstrates this finding with a 2D linear interpolation. The generalization error surface of the target language of XLM-R is much flatter compared to mBERT, perhaps the fundamental reason why XLM-R performs better than mBERT in zero-shot transfer, similar to findings in other computer vision models (Li et al., 2018). As we discuss in Section 7.3, these two findings support our hypothesis that zero-shot cross-lingual transfer is an under-specified optimization problem. As Fig. 7.6 shows, the solution found by zero-shot transfer lies in a non-flat region of target language generalization error surface, causing the high variance of zero-shot transfer on the target language. In contrast, the same solution lies in a flat region of source language generalization error surface, causing the low variance on the source language.

7.6 Discussion

In this chapter, we have presented evidence that zero-shot cross-lingual transfer is an under-specified optimization problem, and the cause of high variance on target language but not the source language tasks during zero-shot cross-lingual transfer. This finding holds across 4 tasks, 2 source languages and 8 target languages. While this chapter focuses on zero-shot cross-lingual transfer, similar high variance has been observed in cross-lingual



(a) EN-HI Parsing with mBERT (b) EN-HI Parsing with XLM-R (c) EN-RU NER with mBERT (d) EN-RU NER with XLM-R



(e) AR-DE POS with mBERT (f) AR-DE POS with XLM-R (g) AR-ZH XNLI with mBERT (h) AR-ZH XNLI with XLM-R

Figure 7.6: Normalized performance of 2D linear interpolation between bilingual model and monolingual models. The x-axis and the y-axis are the α_1 and α_2 in Equation 7.3, respectively. By comparing mBERT and XLM-R, we observe that XLM-R has a flatter target language generalization error surface compared to mBERT. Different language pairs and tasks combination shows similar trends.

transfer with silver data (chapter 8) and few-shot cross-lingual transfer (Zhao et al., 2021), despite outperforming zero-shot cross-lingual transfer. It suggests that they are likely solving a similar under-specified optimization problem, due to the quality of the silver data or the variance of few-shot data selection impacting the gradient direction.

Therefore, addressing this issue may yield significant improvements to zero-shot cross-lingual transfer. Training bigger encoders addresses this issue indirectly by producing encoders with flatter cross-lingual generalization error surfaces. However, a more robust solution may be found by introducing constraints into the optimization problem that directly addresses the under-specification of the optimization.

Silver target data is a potential way to further constrain the optimization problem. Silver target data can be created with machine translation and automatically labeling by either data projection or self training. In chapter 8, we will explore using silver target data to constrain the optimization problem and improve cross-lingual transfer performance.

Similarly, few-shot cross-lingual transfer is a potential way to further constrain the optimization problem. Zhao et al. (2021) finds that few-shot overall improves over zero-shot and it is important to first train on source language then fine-tune with the few-shot target language example. Through the lens of our analysis, this finding is intuitive since fine-tuning with a small amount of target data provides a guidance (gradient direction) to narrow down the solution space, leading to a potentially better solution for the target language. The initial fine-tuning with the source data is also important since it provides a good starting point. Additionally, Zhao et al. (2021) observes that the choice of shots matters. This is expected

as it significantly impacts the quality of the gradient direction.

Unsupervised model selection like Chen and Ritter (2020) and optimization regularization like Aghajanyan et al. (2021) have been proposed in the literature to improve zero-shot cross-lingual transfer. Through the lens of our analysis, both solutions attempt to constrain the optimization problem. As none of the existing techniques fully constrain the optimization, future work should study the combination of existing techniques and develop new techniques on top of it instead of studying one technique at a time.

Chapter 8

Do Data Projection and Self-training Constrain Zero-shot Cross-lingual Transfer Optimization?

8.1 Introduction

In chapter 7, we identify that zero-shot cross-lingual transfer is under-specified optimization. Additionally, performance on the target language in zero-shot cross-lingual transfer is often far below that of within-language supervision, especially in structured prediction tasks (Ruder et al., 2021). To address these challenges, additional constraints need to be added to the optimization problem. One way to achieve this is to add a learning signal for the target language. However, in zero-shot cross-lingual transfer, no target language supervision is available. Thus, we consider data projection and self-training. Before the advent of cross-lingual representations, such as in multilingual word embeddings and mBERT, cross-lingual transfer was approached largely as a data projection problem: one either translated and aligned the source training data to the target language, or at test time one translated target language inputs to the source language for prediction (Yarowsky and Ngai, 2001). Instead of obtaining the label by alignment and projection, we could also obtain the label using the zero-shot model, similar to traditional self-training (Yarowsky, 1995).

We show that by augmenting the source language training data with "silver" data in the target language—either via projection of the source data to the target language or via self-training with translated text—zero-shot performance can be improved, providing constraints to the optimization. Further improvements might come from using better pretrained encoders or improving on a projection strategy through better automatic translation models or better alignment models. In this chapter, we explore all the options above, finding that *everything is all it takes* to best constrain the optimization and achieve our best empirical results,

suggesting that a silver bullet strategy does not currently exist.

Specifically, we evaluate: cross-lingual data projection and self-training techniques with different machine translation and word alignment components, the impact of bilingual and multilingual contextualized encoders on each data projection and self-training component, and the use of different encoders in task-specific models. We also offer suggestions for practitioners operating under different computation budgets on four tasks: event extraction, named entity recognition, part-of-speech tagging, and dependency parsing, following recent work that uses English-to-Arabic tasks as a test bed (Lan et al., 2020). We then apply data projection and self-training to three structured prediction tasks—named entity recognition, part-of-speech tagging, and dependency parsing—in multiple target languages. Additionally, we use self-training as a control against data projection to determine in which situations data projection improves performance.

This chapter is adapted from Yarmohammadi et al. (2021), which is a distillation of the Phase 1 evaluation effort of the BETTER team at Johns Hopkins University and University of Rochester, supported by IARPA BETTER (2019-19051600005). My colleagues on the BETTER team are responsible for the codebase for our BETTER system, and their efforts form the core of this paper. My co-first-author Mahsa Yarmohammadi and I designed and ran most of the experiments reported in this paper and drafted the paper. Marc Marone and Seth Ebner assisted in writing and editing. Other contributions made by my colleagues will be indicated throughout the chapter.

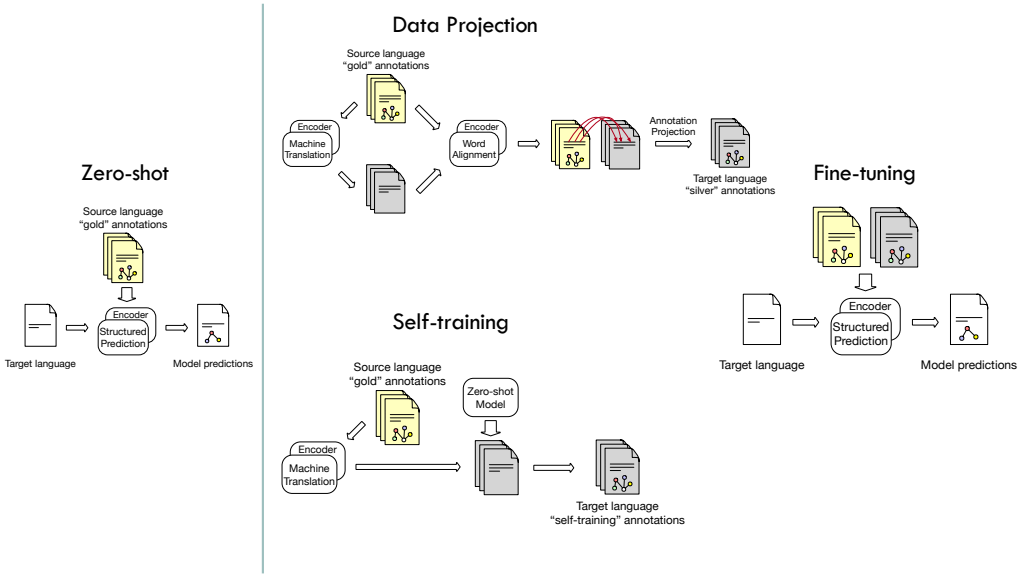


Figure 8.1: Process for creating projected "silver" data from source "gold" data. Downstream models are trained on a combination of gold and silver data. Components in boxes have learned parameters. This figure is made by Seth Ebner and Mahsa Yarmohammadi.

	Base	**Large**
Multilingual	mBERT (Devlin et al.)	XLM-R (Conneau et al.)
Bilingual	GBv4 (Lan et al.)	L64K & L128K **(Ours)**

Table 8.1: Encoders supporting English and Arabic.

8.2 Universal Encoders

While massively multilingual encoders like mBERT and XLM-R enable strong zero-shot cross-lingual performance (Wu and Dredze, 2019; Conneau et al., 2020a), they suffer from the curse of multilinguality (Conneau et al., 2020a): cross-lingual effectiveness suffers as the number of supported languages increases for a fixed model size. We would therefore expect that when restricted to only the source and target languages, a bilingual model should

perform better than (or at least on par with) a multilingual model of the same size, assuming both languages have sufficient corpora (Wu and Dredze, 2020a). If a practitioner is interested in only a small subset of the supported languages, *is the multilingual model still the best option*?

To answer this question, we use English and Arabic as a test bed. In Table 8.1, we summarize existing publicly available encoders that support both English and Arabic.[1] Base models are 12-layer Transformers ($\texttt{d\_model} = 768$), and large models are 24-layer Transformers ($\texttt{d\_model} = 1024$) (Vaswani et al., 2017). As there is no publicly available large English–Arabic bilingual encoder, we train two encoders from scratch, named L64K and L128K, with vocabulary sizes of 64K and 128K, respectively.[2] With these encoders, we can determine the impacts of model size and the number of supported languages.

8.3 Data Projection and Self-Training

We create silver versions of the data by automatically projecting annotations from source English gold data to their corresponding machine translations in the target language or labeling the translations with the zero-shot model. Data projection transfers word-level annotations in a source language to a target language via word-to-word alignments (Yarowsky, Ngai, and Wicentowski, 2001). The technique has been used to create cross-

[1] We do not include multilingual T5 (Xue et al., 2021) as it is still an open question on how to best utilize text-to-text models for structured prediction tasks (Ruder et al., 2021).

[2] L128K available at `https://huggingface.co/jhu-clsp/roberta-large-eng-ara-128k`

lingual datasets for a variety of structured natural language processing tasks, including named entity recognition (Stengel-Eskin et al., 2019) and semantic role labeling (Akbik et al., 2015; Aminian, Rasooli, and Diab, 2017; Fei, Zhang, and Ji, 2020). Labeling data with learned models and using it to further train the model is referred to as self-training (Yarowsky, 1995). This technique has been used for cross-lingual transfer including text classification (Eisenschlos et al., 2019). However, we differ with prior work as we label the translation of source gold data instead of assuming access to unlabeled corpus in target language.

For data projection, to create silver data, as shown in Figure 8.1, we: (1) translate the source text to the target language using the MT system described in Section 8.5.2, (2) obtain word alignments between the original and translated parallel text using a word alignment tool, and (3) project the annotations along the word alignments. We then combine silver target data with gold source data to augment the training set for the structured prediction task. For self-training, the step (1) is shared with data projection, and we use the zero-shot model to label the translation.

For step (1), we rely on a variety of source-to-target MT systems. To potentially leverage monolingual data, as well as contextualized cross-lingual information from pretrained encoders, we feed the outputs of the final layer of frozen pretrained encoders as the inputs to the MT encoders. We consider machine translation systems: (i) whose parameters are randomly initialized, (ii) that incorporate information from massively multilingual encoders, and (iii) that incorporate information from bilingual encoders that have been trained on only

the source and target languages.

After translating source sentences to the target language, in step (2) we obtain a mapping of the source words to the target words using publicly available automatic word alignment tools. Similarly to our MT systems, we incorporate contextual encoders in the word aligner. We hypothesize that better word alignment yields better silver data, and better information extraction consequently.

For step (3), we apply direct projection to transfer labels from source sentences to target sentences according to the word alignments. Each target token receives the label of the source token aligned to it (token-based projection). For multi-token spans, the target span is a contiguous span containing all aligned tokens from the same source span (span-based projection), potentially including tokens not aligned to the source span in the middle. Three of the IE tasks we consider—ACE, named entity recognition, and BETTER—use span-based projection, and we filter out projected target spans that are five times longer than the source spans. Two syntactic tasks—POS tagging and dependency parsing—use token-based projection. For dependency parsing, following Tiedemann, Agić, and Nivre (2014), we adapt the disambiguation of many-to-one mappings by choosing as the head the node that is highest up in the dependency tree. In the case of a non-aligned dependency head, we choose the closest aligned ancestor as the head.

To address issues like translation shift, filtered projection (Akbik et al., 2015; Aminian, Rasooli, and Diab, 2017) has been proposed to obtain higher precision but lower recall projected data. To maintain the same amount of silver data as gold data, in this chapter we

do not use any task-specific filtered projection methods to remove any sentence.

8.4 Tasks

We employ our silver dataset creation approach on a variety of tasks. For English–Arabic experiments, we consider ACE, BETTER, NER, POS tagging, and dependency parsing. For multilingual experiments, we consider NER, POS tagging, and dependency parsing. We use English as the source language and 8 typologically diverse target languages: Arabic, German, Spanish, French, Hindi, Russian, Vietnamese, and Chinese. Because of the high variance of cross-lingual transfer as shown in chapter 6, we report the average test performance of three runs with different predefined random seeds (except for ACE).[3] For model selection and development, we use the English dev set in the zero-shot scenario and the combined English dev and silver dev sets in the silver data scenario. Mahsa Yarmohammadi ran the experiment on English–Arabic ACE and BETTER with the help of Shabnam Behzad.

8.4.1 ACE

Automatic Content Extraction (ACE) 2005 (Walker et al., 2006) provides named entity, relation, and event annotations for English, Chinese, and Arabic. We conduct experiments on English as the source language and Arabic as the target language. We use the OneIE framework (Lin et al., 2020), a joint neural model for information extraction, which has

[3]We report one run for ACE due to long fine-tuning time.

shown state-of-the-art results on all subtasks. We use the same hyperparameters as in Lin et al. (2020) for all of our experiments. We use the OneIE scoring tool to evaluate the prediction of entities, relations, event triggers, event arguments, and argument roles. For English, we use the same English document splits as (Lin et al., 2020). That work does not consider Arabic, so for Arabic we use the document splits from (Lan et al., 2020).

We used the OneIE v0.4.8 codebase[4] with the following hyperparameters: Adam optimizer (Kingma and Ba, 2014) for 60 epochs with a learning rate of 5e-5 and weight decay of 1e-5 for the encoder, and a learning rate of 1e-3 and weight decay of 1e-3 for other parameters. Two-layer feed-forward network with a dropout rate of 0.4 for task-specific classifiers, 150 hidden units for entity and relation extraction, and 600 hidden units for event extraction. β_v and β_e set to 2 and θ set to 10 for global features.

8.4.2 Named Entity Recognition

We use WikiAnn (Pan et al., 2017) for English–Arabic and multilingual experiments. The labeling scheme is BIO with 3 types of named entities: PER, LOC, and ORG. On top of the encoder, we use a linear classification layer with softmax to obtain word-level predictions. The labeling is word-level while the encoders operate at subword-level, thus, we mask the prediction of all subwords except for the first one. We evaluate NER performance by F1 score of the predicted entity.

We use the Adam optimizer with a learning rate of 2e-5 with linear warmup for the first

[4]http://blender.cs.illinois.edu/software/oneie/

10% of total steps and linear decay afterwards, and train for 5 epochs with a batch size of 32. We adapt the same post-processing step as Section 3.2.3 to obtain valid BIO sequences. We set the maximum sequence length to 128 during fine-tuning, and use a sliding window of context to include subwords beyond the first 128. At test time, we use the same maximum sequence length.

8.4.3 Part-of-speech Tagging

We use the Universal Dependencies (UD) Treebank (v2.7; Zeman, 2020b). We use the following treebanks: Arabic-PADT, German-GSD, English-EWT, Spanish-GSD, French-GSD, Hindi-HDTB, Russian-GSD, Vietnamese-VTB, and Chinese-GSD. Similar to NER, we use a word-level linear classifier on top of the encoder, and evaluate performance by the accuracy of predicted POS tags. We use the same fine-tuning hyperparameter and maximum sequence length as NER.

8.4.4 Dependency Parsing

We use the same treebanks as the POS tagging task. For the task-specific layer, we use the graph-based parser of Dozat and Manning (2016), but replace their LSTM encoder with our encoders of interest. We follow the same policy as that in NER for masking non-first subwords. We predict only the universal dependency labels, and we evaluate performance by labeled attachment score (LAS), ignoring punctuations (PUNCT) and symbols (SYM).

We use the same fine-tuning hyperparameter as NER. We set the maximum sequence length to the first 128 subwords during fine-tuning, and the first 128 words at test time.

8.4.5 BETTER

The Better Extraction from Text Towards Enhanced Retrieval (BETTER) Program[5] develops methods for extracting increasingly fine-grained semantic information in a target language, given gold annotations only in English. We focus on the coarsest "Abstract" level, where the goal is to identify events and their agents and patients. The documents come from the news-specific portion of Common Crawl. We report the program-defined "combined F1" metric, which is the product of "event match F1" and "argument match F1", which are based on an alignment of predicted and reference event structures.

To find all events in a sentence and their corresponding arguments, we model the structure of the events as a tree, where event triggers are children of the "virtual root" of the sentence and arguments are children of event triggers (Cai et al., 2018). Each node is associated with a span in the text and is labeled with an event or argument type label.

We use a model for event structure prediction that has three major components: a contextualized encoder, tagger, and typer (Xia et al., 2021).[6] This model is designed by my BETTER team colleague. The tagger is a BiLSTM-CRF BIO tagger (Panchendrarajan and Amaresan, 2018) trained to predict child spans conditioned on parent spans and labels. The typer is a feedforward network whose inputs are a parent span representation, parent label

[5] https://www.iarpa.gov/index.php/research-programs/better

[6] Code available at https://github.com/hiaoxui/span-finder

embedding, and child span representation. The tree is produced level-wise at inference time, first predicting event triggers, typing them, and then predicting arguments conditioned on the typed triggers.

The codebase for event structure prediction uses AllenNLP (Gardner et al., 2018). The contextual encoder produces representations for the tagger and typer modules. Span representations are formed by concatenating the output of a self-attention layer over the span's token embeddings with the embeddings of the first and last tokens of the span. The BiLSTM-CRF tagger has 2 layers, both with hidden size of 2048. We use a dropout rate of 0.3 and maximum sequence length of 512. Child span prediction is conditioned on parent spans and labels, so we represent parent labels with an embedding of size 128. We use Adam optimizer to fine-tune the encoder with a learning rate of 2e-5, and we use a learning rate of 1e-3 for other components. The tagger loss is negative log likelihood and the typer loss is cross entropy. We equally weight both losses and train against their sum. The contextual encoder is not frozen.

8.5 Experiments

8.5.1 Universal Encoders

We train two English–Arabic bilingual encoders. Both of them are 24-layer Transformers ($\texttt{d\_model} = 1024$), the same size as XLM-R large. We use the same Common Crawl

corpus as XLM-R for pretraining. Additionally, we also use English and Arabic Wikipedia, Arabic Gigaword (Parker et al., 2011), Arabic OSCAR (Ortiz Suárez, Romary, and Sagot, 2020), Arabic News Corpus (El-Khair, 2016), and Arabic OSIAN (Zeroual et al., 2019). In total, we train with 9.2B words of Arabic text and 26.8B words of English text, more than either XLM-R (2.9B words/23.6B words) or GBv4 (4.3B words/6.1B words).[7] We build two English–Arabic joint vocabularies using SentencePiece (Kudo and Richardson, 2018), resulting in two encoders: **L64K** and **L128K**. For the latter, we additionally enforce coverage of all Arabic characters after normalization.

We pretrain each encoder with a batch size of 2048 sequences and 512 sequence length for 250K steps from scratch,[8] roughly 1/24 the amount of pretraining compute of XLM-R. Training takes 8 RTX 6000 GPUs roughly three weeks. We follow the pretraining recipe of RoBERTa (Liu et al., 2019b) and XLM-R. We omit the next sentence prediction task and use a learning rate of 2e-4, Adam optimizer, and linear warmup of 10K steps then decay linearly to 0, multilingual sampling alpha of 0.3, and the fairseq (Ott et al., 2019) implementation.

8.5.2 Machine Translation

The machined translation component is developed by Haoran Xu and Kenton Murray, initially developed for the BETTER program and later improved by Xu, Van Durme, and Murray (2021). The detailed description of this component can be found in Yarmohammadi

[7] We measure word count with `wc -w`.

[8] While we use XLM-R as the initialization of the Transformer, due to vocabulary differences, the learning curve is similar to that of pretraining from scratch.

Encoder	BLEU
Public	12.7
None	14.9
mBERT	15.7
GBv4	15.7
XLM-R	16.0
L64K	**16.2**
L128K	15.8

Table 8.2: BLEU scores of MT systems with different pre-trained encoders on English–Arabic IWSLT'17.

et al. (2021). In summary, it uses the final contextual embeddings from a frozen bilingual or multilingual encoder as the input of the MT encoder, instead of a randomly initialized embedding matrix ("None"). We also include a publicly released model ("public") that has been demonstrated to perform well (Tiedemann, 2020).[9] Table 8.2 shows the denormalized and detokenized BLEU scores for English–Arabic MT systems with different encoders on the IWLST'17 test set using sacreBLEU (Post, 2018). The use of contextualized embeddings from pretrained encoders results in better performance than using a standard randomly initialized MT model regardless of which encoder is used. The best performing system uses our bilingual L64K encoder, but all pretrained encoder-based systems perform well and within 0.5 BLEU points of each other.

[9]The public MT model is available at `https://huggingface.co/Helsinki-NLP/opus-mt-en-ar`

Model	**Layer†**	**AER**	**P**	**R**	**F**
fast-align*	n/a	47.4	53.9	51.4	52.6
Awesome-align w/o FT					
mBERT	8	35.6	78.5	54.5	64.4
GBv4	8	**32.7**	**85.6**	55.4	**67.3**
XLM-R	16	40.1	78.6	48.4	59.9
L64K	17	34.0	81.5	**55.5**	66.0
L128K	17	35.1	80.0	54.5	64.9
Awesome-align w/ FT					
mBERT$_{ft}$	8	30.0	81.9	**61.2**	70.0
GBv4$_{ft}$	8	29.3	86.9	59.7	70.7
XLM-R$_{ft}$	18	**27.8**	**90.3**	60.2	**72.2**
L64K$_{ft}$	17	29.1	84.9	60.9	70.9
L128K$_{ft}$	16	32.2	80.3	58.7	67.8
Awesome-align w/ FT & supervision					
XLM-R$_{ft.s}$	16	**23.3**	92.5	**65.6**	**76.7**
L128K$_{ft.s}$	17	23.5	**93.7**	64.6	76.5

Table 8.3: Alignment performance on GALE EN–AR. *Trained on MT bitext. †We report the best layer of each encoder based on dev alignment error rate (AER).

8.5.3 Word Alignment

Until recently, alignments have typically been obtained using unsupervised statistical models such as GIZA++ (Och and Ney, 2003) and fast-align (Dyer, Chahuneau, and Smith, 2013). Recent work has focused on using the similarities between contextualized embeddings to obtain alignments (Jalili Sabet et al., 2020; Daza and Frank, 2020; Dou and Neubig, 2021), achieving state-of-the-art performance.

We use two automatic word alignment tools: fast-align, a widely used statistical alignment tool based on IBM models (Brown et al., 1993); and Awesome-align (Dou and Neubig,

2021), a contextualized embedding-based word aligner that extracts word alignments based on similarities of the tokens' contextualized embeddings. Awesome-align achieves state-of-the-art performance on five language pairs. Optionally, Awesome-align can be fine-tuned on parallel text with objectives suitable for word alignment and on gold alignment data. Mahsa Yarmohammadi ran the experiment on word alignment with the help of Shabnam Behzad, while I adapted the code to allow using any encoder.

We benchmark the word aligners on the gold standard alignments in the GALE Arabic–English Parallel Aligned Treebank (Li et al., 2012). We use the same data splits as Stengel-Eskin et al. (2019), containing 1687, 299, and 315 sentence pairs in the train, dev, and test splits, respectively. To obtain alignments using fast-align, we append the test data to the MT training bitext and run the tool from scratch. Awesome-align extracts the alignments for the test set based on pretrained contextualized embeddings. These encoders can be fine-tuned using the parallel text in the train and dev sets. Additionally, the encoders can be further fine-tuned using supervision from gold word alignments.

8.5.3.1 Intrinsic Evaluation

Table 8.3 shows the performance of word alignment methods on the GALE English–Arabic alignment dataset. Awesome-align outperforms fast-align, and fine-tuned Awesome-align (*ft*) outperforms models that were not fine-tuned. Incorporating supervision from the gold alignments (*s*) leads to the best performance.

8.6 Cross-lingual Transfer

One might optimistically consider that the latest multilingual encoder (in this case XLM-R) in the zero-shot setting would achieve the best possible performance, which suggest data projection or self-training could not constrain the zero-shot cross-lingual optimization. However, in our extensive experiments in Table 8.4 and Table 8.5, we find that data projection and self-training could provide useful constraints and improve over zero-shot approach. In this section, we explore the impact of each factor within the silver data creation process.

8.6.1 English–Arabic Experiments

In Table 8.4, we present the Arabic test performance of five tasks under all combinations considered. The "MT" and "Align" columns indicate the models used for the translation and word alignment components of the silver data creation process. For ACE, we report results on the average of six metrics. [10] For a large bilingual encoder, we use L128K instead of L64K due to its slightly better performance on English ACE.

8.6.1.1 Impact of Data Projection

By comparing any group against group Z, we observe adding silver data yields better or equal performance to zero-shot in at least some setup in the IE tasks (ACE, NER, and BETTER). For syntax-related tasks, we observe similar trends, with the exception of XLM-

[10] Six metrics include entity, relation, trigger identification and classification, and argument identification and classification accuracies.

	MT	Align	ACE	NER	POS	Parsing	BET.	ACE	NER	POS	Parsing	BET.
			mBERT (base, multilingual)					*XLM-R (large, multilingual)*				
(Z)	-	-	27.0	41.6	59.7	29.2	39.9	**45.1**	46.4	73.3	**48.0**	50.8
(A)	public	FA	+2.5	-3.8	+8.5	+7.3	+2.6	-7.5	-0.1	-7.7	-9.5	-1.6
(B)	public	mBERT	+6.5	+0.2	+8.5	+7.6	+2.3	-4.4	+6.9	-6.1	-8.4	-2.6
(B)	public	XLM-R	+0.9	-2.9	+9.5	+9.0	-1.2	-10.0	+0.0	-5.9	-8.8	-6.3
(C)	public	mBERT$_{ft}$	+7.8	**+5.6**	+7.7	+10.0	+4.1	-0.6	+7.4	-8.0	-6.8	+0.3
(C)	public	XLM-R$_{ft}$	+7.7	+4.9	+6.2	+9.3	+4.5	-2.6	+7.0	-9.0	-7.6	+1.0
(C)	public	XLM-R$_{ft.s}$	+7.3	+1.5	+10.1	**+12.4**	+4.8	-3.0	+9.1	-3.8	-3.7	**+2.3**
(D)	public	GBv4$_{ft}$	+8.5	+4.3	+5.9	+8.9	+5.0	-1.5	+7.7	-9.4	-9.1	-0.1
(D)	public	L128K$_{ft}$	+6.4	+3.1	+6.5	+8.2	+1.6	-1.6	+6.1	-9.0	-9.4	-3.6
(D)	public	L128K$_{ft.s}$	+7.0	+3.7	**+10.3**	+11.8	**+5.4**	-0.3	+5.2	-4.4	-4.6	+2.1
(E)	GBv4	mBERT$_{ft}$	+8.4	+3.2	+7.7	+9.9	+4.7	-1.5	+3.2	-7.1	-6.7	+0.7
(E)	GBv4	XLM-R$_{ft}$	+9.6	+1.8	+7.0	+9.5	+5.2	-0.4	+1.4	-8.3	-7.7	+1.4
(E)	L128K	mBERT$_{ft}$	**+12.1**	+3.3	+7.9	+9.9	+4.7	-1.4	+7.2	-8.1	-6.7	+1.3
(E)	L128K	XLM-R$_{ft}$	+10.2	-1.9	+6.1	+9.4	+4.8	-0.5	+4.6	-9.8	-7.5	+2.0
(S)	public	ST	-	+5.5	+0.1	-20.3	+0.3	-	**+10.0**	**+1.8**	-29.6	+1.2
			GBv4 (base, bilingual)					*L128K (large, bilingual)*				
(Z)	-	-	46.0	45.4	64.7	33.2	41.7	42.7	46.3	67.9	36.7	40.9
(C)	public	mBERT$_{ft}$	+0.6	+3.7	+2.6	+6.9	+7.5	+2.7	+8.2	-0.9	+4.9	+11.7
(C)	public	XLM-R$_{ft}$	-1.4	**+4.5**	+1.8	+6.0	+8.4	+1.2	**+9.0**	-2.5	+3.9	+10.5
(C)	public	XLM-R$_{ft.s}$	-0.1	+3.4	**+5.1**	**+9.2**	+8.0	+2.7	+7.0	+1.2	**+7.2**	**+12.1**
(E)	GBv4	mBERT$_{ft}$	-0.1	+0.1	+3.3	+7.2	+8.1	+4.2	-0.5	-0.1	+5.1	+11.2
(E)	GBv4	XLM-R$_{ft}$	+0.1	+0.4	+1.5	+6.0	**+9.7**	+2.4	+0.0	-1.3	+4.2	+10.8
(E)	L128K	mBERT$_{ft}$	-0.6	+1.0	+2.6	+6.1	+7.4	**+5.5**	+0.8	-0.7	+4.7	+10.6
(E)	L128K	XLM-R$_{ft}$	+0.9	-2.1	+1.1	+5.5	+7.8	+4.4	-3.6	-2.2	+4.1	+11.3
(F)	GBv4	GBv4$_{ft}$	+0.0	-1.9	+1.6	+4.5	+9.1	+2.0	-0.3	-1.7	+3.2	+10.9
(F)	GBv4	L128K$_{ft}$	-0.9	-1.4	+1.5	+4.1	+5.7	+2.3	-1.7	-2.4	+2.6	+8.3
(F)	L128K	GBv4$_{ft}$	-4.3	-1.0	+0.4	+4.1	+7.4	+4.1	-3.6	-2.1	+2.3	+11.4
(F)	L128K	L128K$_{ft}$	-3.5	-1.1	+0.3	+3.8	+ 4.5	+2.9	+0.1	-2.9	+2.0	+6.7
(F)	L128K	L128K$_{ft.s}$	**+1.9**	+0.2	+3.3	+7.4	+7.2	+2.8	-1.8	+0.8	+6.0	+11.8
(S)	public	ST	-	-2.5	-1.3	-18.6	+1.9	-	+7.1	**+1.5**	-21.7	+8.1

Table 8.4: Performance of Arabic on 5 tasks under various setups. Cells are colored by performance difference over zero-shot baseline: +5 or more, +1 to +5, -1 to -5, -5 or more. **Highlights** indicate the best setting for each task (best viewed in color). The best setting for each task and encoder combination is **bolded**. We order four encoders along two axes, similar to Table 8.1.

R. We hypothesize that XLM-R provides better syntactic cues than those obtainable from the alignment, which we discuss later in relation to self-training.

8.6.1.2 Impact of Word Aligner

By comparing groups A, B, and C of the same encoder, we observe that Awesome-align performs overall better than statistical MT-based fast-align (FA). Additional fine-tuning (*ft*) on MT training bitext further improves its performance. As a result, we use fine-tuned aligners for further experiments. Moreover, incorporating supervised signals from gold alignments in the word alignment component (*ft.s*) often helps performance of the task. In terms of computation budget, these three groups use a publicly available MT system ("public"; Tiedemann, 2020) and require only fine-tuning the encoder for alignment, which requires small additional computation.

8.6.1.3 Impact of Encoder Size

Large bilingual or multilingual encoders tend to perform better than base encoders in the zero-shot scenario, with the exception of the bilingual encoders on ACE and BETTER. While we observe base size encoders benefit from reducing the number of supported languages (from 100 to 2), for large size encoders trained much longer, the zero-shot performance of the bilingual model is worse than that of the multilingual model. After adding silver data from group C based on the public MT model and the fine-tuned aligner, the performance gap between base and large models tends to shrink, with the exception of both bilingual and

multilingual encoders on NER. In terms of computation budget, training a bilingual encoder requires significant additional computation.

8.6.1.4 Impact of Encoder on Word Aligner

By comparing groups C and D (in multilingual encoders) or groups E and F (in bilingual encoders), we observe bilingual encoders tend to perform slightly worse than multilingual encoders for word alignment. If bilingual encoders exist, using them in aligners requires little additional computation.

8.6.1.5 Impact of Encoder on MT

By comparing groups C and E, we observe the performance difference between the bilingual encoder based MT and the public MT depends on the task and encoder, and neither MT system clearly outperforms the other in all settings, despite the bilingual encoder having a better BLEU score. The results suggest that both options should be explored if one's budget allows. In terms of computation budget, using pretrained encoders in a custom MT system requires medium additional computation.

8.6.1.6 Impact of Label Source

To assess the quality of the projected annotations in the silver data, we consider a different way to automatically label translated sentences: self-training (ST; Yarowsky, 1995). For self-training, we translate the source data to the target language, label the translated data

using a zero-shot model trained on source data, and combine the labeled translations with the source data to train a new model.[11] Compared to the silver data, the self-training data has the same underlying text but a different label source.

We first observe that self-training for parsing leads to significantly worse performance due to the low quality of the predicted trees. By comparing groups S and C, which use the same underlying text, we observe that data projection tends to perform better than self-training, with the exceptions of POS tagging with a large encoder and NER with a large multilingual encoder. These results suggest that the external knowledge[12] in the silver data complements the knowledge obtainable when the model is trained with source language data alone, but when the zero-shot model is already quite good (like for POS tagging) data projection can harm performance compared to self-training.

8.6.2 Multilingual Experiments

In Table 8.5, we present the test performance of three tasks for eight target languages. We use the public MT system (Tiedemann, 2020) and non-fine-tuned Awesome-align with mBERT as the word aligner for data projection—a setup with the smallest computation budget—due to computation constraints. We consider both data projection (+Proj) and self-training (+Self). We use silver data in addition to English gold data for training. We use

[11] This setup differs from traditional zero-shot self-training in cross-lingual transfer, as the traditional setup assumes unlabeled corpora in the target language(s) (Eisenschlos et al., 2019) instead of translations of the source language data.

[12] "External knowledge" refers to knowledge introduced into the downstream model as a consequence of the particular decisions made by the aligner (and subsequent projection).

Encoder	Data	ar	de	en	es	fr	hi	ru	vi	zh	Average
NER (F1)											
mBERT	Zero-shot	41.6	**78.8**	83.9	73.1	79.5	66.2	63.4	70.8	51.8	67.7
	+ Self	**+7.7**	-0.5	**+0.4**	+4.8	**+2.4**	-2.5	**+2.7**	**+1.2**	+1.4	**+2.0**
	+ Proj	-5.8	-0.6	+0.3	+3.6	+0.2	**+0.4**	-1.7	-2.0	**+2.3**	-0.4
	+ Proj (Bi)	+0.3	-0.7	+0.1	**+5.2**	-0.6	-2.1	-1.1	+0.3	+0.0	+0.2
XLM-R	Zero-shot	46.4	79.5	83.9	76.1	80.0	70.9	**70.5**	77.0	40.2	69.4
	+ Self	**+11.2**	**+0.9**	**+0.6**	**+1.0**	**+0.5**	+2.1	-1.5	**+1.7**	+2.3	**+2.1**
	+ Proj	+1.7	-0.7	-0.1	-3.9	-1.2	+1.2	-4.8	-9.1	+14.2	-0.3
	+ Proj (Bi)	+6.9	+0.4	-0.2	-4.3	-1.5	**+3.2**	-3.3	-5.2	**+15.1**	+1.2
POS (ACC)											
mBERT	Zero-shot	59.7	89.6	**96.9**	87.5	88.7	69.5	81.9	62.6	66.6	78.1
	+ Self	+0.3	**+0.5**	+0.0	**+0.4**	**+0.4**	-0.3	**+0.5**	**+0.4**	**+1.7**	**+0.4**
	+ Proj	+6.9	-3.2	+0.0	-3.8	-3.9	+1.3	-6.6	-7.4	-4.1	-2.3
	+ Proj (Bi)	**+8.5**	-2.6	-0.1	-3.2	-3.0	**+1.6**	-5.7	-6.9	-3.9	-1.7
XLM-R	Zero-shot	73.3	**91.5**	**98.0**	**89.3**	**90.0**	78.6	86.8	**65.2**	53.6	80.7
	+ Self	**+1.6**	-0.3	+0.0	+0.0	+0.0	**+2.0**	**+0.1**	-0.4	**+11.7**	**+1.6**
	+ Proj	-7.1	-5.4	-0.5	-6.3	-5.9	-6.0	-10.5	-8.9	+9.7	-4.6
	+ Proj (Bi)	-6.1	-4.6	-0.1	-4.9	-4.6	-5.5	-10.4	-8.7	+9.4	-4.0
Parsing (LAS)											
mBERT	Zero-shot	29.2	**67.7**	79.7	**68.9**	**73.2**	31.2	**60.6**	**33.6**	**29.4**	**52.6**
	+ Self	-20.6	-34.2	+0.1	-41.6	-41.1	-15.3	-35.2	-17.8	-14.5	-24.5
	+ Proj	**+9.1**	-2.1	**+1.1**	-4.9	-5.8	**+6.0**	-5.6	-7.2	-2.1	-1.3
	+ Proj (Bi)	+7.6	-1.6	+0.5	-3.8	-4.5	+5.7	-4.8	-7.2	-2.5	-1.2
XLM-R	Zero-shot	**48.0**	**69.6**	82.6	**73.6**	**76.1**	**43.1**	**70.3**	**38.4**	15.0	**57.4**
	+ Self	-30.4	-29.4	**+0.1**	-39.9	-40.0	-18.3	-33.9	-16.1	-9.7	-24.2
	+ Proj	-8.5	-4.3	+0.0	-10.3	-10.1	-5.7	-14.8	-11.1	+14.5	-5.6
	+ Proj (Bi)	-8.4	-1.6	**+0.1**	-7.7	-7.4	-3.1	-12.7	-9.8	**+15.1**	-3.9

Table 8.5: Performance of NER, POS, and parsing for eight target languages. We use the same color code as Table 8.4.

multilingual training with +Self and +Proj, and bilingual training with +Proj (Bi).

We observe that data projection (+Proj (Bi)) sometimes benefits languages with the lowest zero-shot performance (Arabic, Hindi, and Chinese), with the notable exception of XLM-R on syntax-based tasks (excluding Chinese). For languages closely related to English, data projection tends to hurt performance. We observe that for data projection, training multiple bilingual models (+Proj (Bi)) outperforms joint multilingual training (+Proj). This could be the result of noise from alignments of various quality mutually interfering. In fact, self-training with the same translated text (+Self) outperforms data projection and zero-shot scenarios, again with the exception of parsing. As data projection and self-training use the same translated text and differ only by label source, the results indicate that the external knowledge from frozen mBERT-based alignment is worse than what the model learns from source language data alone. Thus, further performance improvement could be achieved with an improved aligner.

8.7 Related Work

Although projected data may be of lower quality than the original source data due to errors in translation or alignment, it is useful for tasks such as semantic role labeling (Akbik et al., 2015; Aminian, Rasooli, and Diab, 2019), information extraction (Riloff, Schafer, and Yarowsky, 2002), POS tagging (Yarowsky and Ngai, 2001), and dependency parsing (Ozaki et al., 2021). The intuition is that although the projected data may be noisy, training on it

gives a model useful information about the statistics of the target language.

Akbik et al. (2015) and Aminian, Rasooli, and Diab (2017) use bootstrapping algorithms to iteratively construct projected datasets for semantic role labeling. Akbik et al. (2015) additionally use manually defined filters to maintain high data quality, which results in a projected dataset that has low recall with respect to the source corpus. Fei, Zhang, and Ji (2020) and Daza and Frank (2020) find that a non-bootstrapped approach works well for cross-lingual SRL. Advances in translation and alignment quality allow us to avoid bootstrapping while still constructing projected data that is useful for downstream tasks.

Fei, Zhang, and Ji (2020) and Daza and Frank (2020) also find improvements when training on a mixture of gold source language data and projected silver target language data. The intuition of using both gold and projected silver data is to allow the model to see high quality gold data as well as data with target language statistics. Ideas from domain adaptation can be used to make more effective use of gold and silver data to mitigate the effects of language shift (Xu et al., 2021).

8.8 Discussion

In this chapter, we explore the use of silver data via data projection or self-training to constrain the zero-shot cross-lingual transfer optimization, facilitated by neural machine translation and word alignment. Recent advances in pretrained encoders have improved machine translation systems and word aligners in terms of intrinsic evaluation. We conduct

an extensive extrinsic evaluation and study how the encoders themselves—and components containing them—impact performance on a range of downstream tasks and languages.

With a test bed of English–Arabic IE tasks, we find that adding projected silver training data overall yields improvements over zero-shot learning. Comparisons of how each factor in the data projection process impacts performance show that while one might hope for the existence of a silver bullet strategy, the best setup is usually task dependent. In multilingual experiments, we find that silver data tends to help languages with the weakest zero-shot performance, and that it is best used separately for each desired language pair instead of in joint multilingual training.

We also examine self-training with translated text to assess when data projection helps cross-lingual transfer, and find it to be another viable option for obtaining labels for some tasks. Further directions include how to improve alignment quality and how to combine data projection and self-training techniques.

As we observe in this chapter, the best setup to constrain the optimization for each task is task-specific. Thus, to further constrain the optimization and to produce the best performance with the existing encoders, we should consider a bag of techniques mentioned in this chapter and Section 7.6, depending on the computation budget.

Chapter 9

Conclusions

9.1 Contributions

In this thesis, we have attempted to answer a set of questions raised by the surprising cross-lingual effectiveness of Multilingual BERT. By understanding these models through analysis, we have identified ways to improve its cross-lingual representation. In chapter 2, we review the progress of representation learning in NLP and its impact on cross-lingual transfer, and observe that cross-lingual transfer performance improves as better representation learning techniques are developed. With the release of Multilingual BERT (mBERT), in chapter 3, we show that surprisingly mBERT learns cross-lingual representation even without explicit cross-lingual signal, and probes the released model to gain more insight. In chapter 4, we conduct an ablation study on these multilingual models, demonstrating that parameter sharing of transformer contribute the most to the learning of cross-lingual representation, and show monolingual BERT of different language are similar to each other. In chapter 5, we find that mBERT does not learn high quality representation for its lower resource languages, despite trying its best, as monolingual BERT or bilingual BERT paired with similar high resource language performs worse that mBERT for lower resource languages. In chapter 6, we propose two methods for injecting different cross-lingual signal—bilingual dictionary and bitext—into these models, and show that while despite improvement on cross-lingual representation, it is eclipsed by the improvement of scaling up the model. In chapter 7, we show that zero-shot cross-lingual transfer is under-specified optimization, causing its high variance on target languages and much lower variance on source language. To address this issue, constraints need to be introduced into the optimization. Thus, in chapter 8, we

consider using silver target data—created automatically with machine translation based on supervision in source language—to constrain the optimization. We show that it indeed improves zero-shot cross-lingual transfer, despite the best setup of data creation pipeline with encoders is task specific.

9.2 Future Works

9.2.1 Continue Scaling of Multilingual Encoders

As we observe in chapter 6, while injecting cross-lingual signal explicitly into the model helps improve cross-lingual representation for smaller model, simply scaling up the model capacity and data size produces much better representation, as evidenced by Xue et al. (2021) and Goyal et al. (2021). In an orthogonal direction, as we discuss in chapter 5, improving the sample-efficiency of BERT pretraining objective likely leads to better representation, as evidenced by Clark et al. (2020) and Chi et al. (2021b).

However, in this direction, there is one important open question: *Is there a limit of scaling up model size?* In other words, Is there a peak model size, beyond which we will get the same or even worse cross-lingual representation? Kaplan et al. (2020) study the scaling law of monolingual language model, and observe that the cross-entropy loss of language model scales as a power-law with model size and data size. However, it is unclear that whether low cross-entropy loss in this scale translate to better cross-lingual representation.

Chi et al. (2021a) argues that the cross-lingual representation of pretrained multilingual models is the product of the bottleneck effect, which suggests a scenario where the model learn each language well with low cross-entropy loss but fail to align the representation in the cross-lingual fashion. As we observe in chapter 4, such natural alignment is the key to the success of models like mBERT.

We hypothesize that this scenario might exist in theory but not in practise for model sharing all parameters across languages. As the model is trained with gradient descent mixing sentences of different languages, we will not discover such a solution with optimization. Additionally, scaling up model size is much slower compared to scaling up data size, as new data is continuously produced on the Web. Realistically, it is unlikely that we will hit such a limit any time soon even if it exists.

Looking at the current literature, we still observe better cross-lingual representation by scaling up to 10B parameters (Xue et al., 2021; Goyal et al., 2021). There are models like GPT-3 (Brown et al., 2020) with 100B+ parameters or GShard (Lepikhin et al., 2021) and Switch Transformer (Fedus, Zoph, and Shazeer, 2021) with 1T+ parameters. Notably, these models are decoder-only or encoder-decoder based models. While the exact cross-lingual capability of these models are unclear at the moment, as these models are not open sourced. There is early evidence suggesting that these models might process similar cross-lingual capability (Winata et al., 2021). Thus, a further model size scaling of 100x is possible with the current techniques. However, such scaling requires answering new questions.

Can we develop a more scalable architecture and algorithm without sacrificing the

natural alignment? Scaling to 1T (Lepikhin et al., 2021; Fedus, Zoph, and Shazeer, 2021) rely heavily on sparse network or conditional computation—activating certain component of the network depending on the input—with mixture of experts (Jacobs et al., 1991; Jordan and Jacobs, 1994; Shazeer et al., 2017). However, it might harm the natural alignment across languages, as less parameter is shared across all languages compared to dense network. Thus, the model might learns better representation for each language, but worse cross-lingual representation in comparison. To address this challenge, possible directions include continue improvement of mixture of experts, Transformers, and more sample-efficient pretraining algorithm.

Can we efficiently adapt a large pretrained model to a task? Such large models introduce challenges for fine-tuning. GPT-3 (Brown et al., 2020), a model with 175B parameters, addresses this challenge by relying on context-based few-shot learning, showing that with a prompt and some examples, the model can perform new tasks without fine-tuning. However, there is still much room for improvement. This is an active research area with new techniques like prompt fine-tuning (Li and Liang, 2021; Qin and Eisner, 2021; Lester, Al-Rfou, and Constant, 2021). Liu et al. (2021a) survey the ongoing research.

Can we distill knowledge from a large pretrained model for deployment? Large models introduce significant challenges for deployment. We could distill the knowledge from large pretrained models to smaller models to reduce inference time. It includes technique like knowledge-distillation (Hinton, Vinyals, and Dean, 2015)—transferring logits of large models on target languages (soft label)—and self-training (Yarowsky, 1995) or semi-supervised

learning—transferring the prediction of large models on target languages (hard label). As we show in chapter 8, self-training with the same model improves the cross-lingual transfer except for parsing. Additionally, the large model could be potentially pruned or quantized.

9.2.2 Multilingual Multi-modals Models

While this thesis mainly focuses on multilingual models for text, the lessons we learned may transfer to multilingual multi-modals models, e.g. speech, text + speech, text + image, text + video, and text + code. Similar to modern NLP systems, multi-modals systems usually need to support more than one language. Similar to text, we do not have the same amount of supervision for all languages. Thus, it is beneficial to consider the multilingual approach that enable cross-lingual transfer. The main research question in this direction is that how well do lessons we learn from text encoder transfer to multi-modals models? Does cross-lingual representation emerge in multilingual multi-modals models?

Bibliography

Aghajanyan, Armen, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta (2021). "Better Fine-Tuning by Reducing Representational Collapse". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=OQ08SN70M1V.

Aharoni, Roee, Melvin Johnson, and Orhan Firat (June 2019). "Massively Multilingual Neural Machine Translation". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 3874–3884. URL: https://aclanthology.org/N19-1388.

Ahmad, Wasi, Zhisong Zhang, Xuezhe Ma, Eduard Hovy, Kai-Wei Chang, and Nanyun Peng (June 2019). "On Difficulties of Cross-Lingual Transfer with Order Differences: A Case Study on Dependency Parsing". 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)*. Minneapolis, Minnesota: Association

for Computational Linguistics, pp. 2440–2452. URL: https://aclanthology.org/N19-1253.

AI2 (2020). "112 - Alignment of Multilingual Contextual Representations, with Steven Cao". In: *NLP Highlights Podcast*. URL: https://soundcloud.com/nlp-highlights/112-alignment-of-multilingual-contextual-representatio with-steven-cao.

Akbik, Alan, Laura Chiticariu, Marina Danilevsky, Yunyao Li, Shivakumar Vaithyanathan, and Huaiyu Zhu (July 2015). "Generating High Quality Proposition Banks for Multilingual Semantic Role Labeling". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, pp. 397–407. URL: https://aclanthology.org/P15-1039.

Aminian, Maryam, Mohammad Sadegh Rasooli, and Mona Diab (Nov. 2017). "Transferring Semantic Roles Using Translation and Syntactic Information". In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, pp. 13–19. URL: https://aclanthology.org/I17-2003.

Aminian, Maryam, Mohammad Sadegh Rasooli, and Mona Diab (May 2019). "Cross-Lingual Transfer of Semantic Roles: From Raw Text to Semantic Roles". In: *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*.

Gothenburg, Sweden: Association for Computational Linguistics, pp. 200–210. URL: https://aclanthology.org/W19-0417.

Ammar, Waleed, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A Smith (2016). "Massively multilingual word embeddings". In: *arXiv preprint arXiv:1602.01925*.

Artetxe, Mikel, Gorka Labaka, and Eneko Agirre (July 2017). "Learning bilingual word embeddings with (almost) no bilingual data". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 451–462. URL: https://aclanthology.org/P17-1042.

Artetxe, Mikel, Sebastian Ruder, and Dani Yogatama (July 2020). "On the Cross-lingual Transferability of Monolingual Representations". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 4623–4637. URL: https://aclanthology.org/2020.acl-main.421.

Artetxe, Mikel and Holger Schwenk (Mar. 2019). "Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond". In: *Transactions of the Association for Computational Linguistics* 7, pp. 597–610. URL: https://aclanthology.org/Q19-1038.

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E Hinton (2016). "Layer normalization". In: *arXiv preprint arXiv:1607.06450*.

BIBLIOGRAPHY

Baevski, Alexei, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli (Nov. 2019). "Cloze-driven Pretraining of Self-attention Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 5360–5369. URL: https://aclanthology.org/D19-1539.

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Janvin (2003). "A neural probabilistic language model". In: *The journal of machine learning research* 3, pp. 1137–1155.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov (2017). "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146. URL: https://aclanthology.org/Q17-1010.

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer (1993). "The Mathematics of Statistical Machine Translation: Parameter Estimation". In: *Computational Linguistics* 19.2, pp. 263–311. URL: https://aclanthology.org/J93-2003.

Brown, Peter F., Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer (1992). "Class-Based *n*-gram Models of Natural Language". In: *Computational Linguistics* 18.4, pp. 467–480. URL: https://aclanthology.org/J92-4003.

BIBLIOGRAPHY

Brown, Tom B, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. (2020). "Language models are few-shot learners". In: *arXiv preprint arXiv:2005.14165*.

Cai, Jiaxun, Shexia He, Zuchao Li, and Hai Zhao (Aug. 2018). "A Full End-to-End Semantic Role Labeler, Syntactic-agnostic Over Syntactic-aware?" In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 2753–2765. URL: https://aclanthology.org/C18-1233.

Cao, Steven, Nikita Kitaev, and Dan Klein (2020). "Multilingual Alignment of Contextual Word Representations". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=r1xCMyBtPS.

Chang, Pi-Chuan, Michel Galley, and Christopher D. Manning (June 2008). "Optimizing Chinese Word Segmentation for Machine Translation Performance". In: *Proceedings of the Third Workshop on Statistical Machine Translation*. Columbus, Ohio: Association for Computational Linguistics, pp. 224–232. URL: https://aclanthology.org/W08-0336.

Chau, Ethan C., Lucy H. Lin, and Noah A. Smith (Nov. 2020). "Parsing with Multilingual BERT, a Small Corpus, and a Small Treebank". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 1324–1334. URL: https://aclanthology.org/2020.findings-emnlp.118.

BIBLIOGRAPHY

Che, Wanxiang, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu (Oct. 2018). "Towards Better UD Parsing: Deep Contextualized Word Embeddings, Ensemble, and Treebank Concatenation". In: *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Brussels, Belgium: Association for Computational Linguistics, pp. 55–64. URL: https://aclanthology.org/K18-2005.

Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton (2020). "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PMLR, pp. 1597–1607.

Chen, Yang and Alan Ritter (2020). "Model Selection for Cross-Lingual Transfer using a Learned Scoring Function". In: *arXiv preprint arXiv:2010.06127*.

Chi, Zewen, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou (June 2021a). "InfoXLM: An Information-Theoretic Framework for Cross-Lingual Language Model Pre-Training". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 3576–3588. URL: https://aclanthology.org/2021.naacl-main.280.

Chi, Zewen, Shaohan Huang, Li Dong, Shuming Ma, Saksham Singhal, Payal Bajaj, Xia Song, and Furu Wei (2021b). "Xlm-e: Cross-lingual language model pre-training via electra". In: *arXiv preprint arXiv:2106.16138*.

BIBLIOGRAPHY

Clark, Kevin, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning (2020). "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=r1xMH1BtvB.

Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov (July 2020a). "Unsupervised Cross-lingual Representation Learning at Scale". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 8440–8451. URL: https://aclanthology.org/2020.acl-main.747.

Conneau, Alexis and Guillaume Lample (2019). "Cross-lingual Language Model Pretraining". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2019/file/c04c19c2c2474dbf5f7ac4372c5b9af1-Paper.pdf.

Conneau, Alexis, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov (2018). "XNLI: Evaluating Cross-lingual Sentence Representations". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 2475–2485. URL: https://aclanthology.org/D18-1269.

BIBLIOGRAPHY

Conneau, Alexis, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov (July 2020b). "Emerging Cross-lingual Structure in Pretrained Language Models". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 6022–6034. URL: https://aclanthology.org/2020.acl-main.536.

Daza, Angel and Anette Frank (Nov. 2020). "X-SRL: A Parallel Cross-Lingual Semantic Role Labeling Dataset". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 3904–3914. URL: https://aclanthology.org/2020.emnlp-main.321.

Deerwester, Scott, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman (1990). "Indexing by latent semantic analysis". In: *Journal of the American society for information science* 41.6, pp. 391–407.

Delobelle, Pieter, Thomas Winters, and Bettina Berendt (Nov. 2020). "RobBERT: a Dutch RoBERTa-based Language Model". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 3255–3265. URL: https://aclanthology.org/2020.findings-emnlp.292.

Devlin, Jacob (2018). *Multilingual BERT README document*. URL: https://github.com/google-research/bert/blob/a9ba4b8d7704c1ae18d1b28c56c0430d414(multilingual.md (visited on 12/03/2018).

BIBLIOGRAPHY

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (June 2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. URL: https://aclanthology.org/N19-1423.

Dodge, Jesse, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith (2020). "Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping". In: *arXiv preprint arXiv:2002.06305*.

Dou, Zi-Yi and Graham Neubig (Apr. 2021). "Word Alignment by Fine-tuning Embeddings on Parallel Corpora". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, pp. 2112–2128. URL: https://aclanthology.org/2021.eacl-main.181.

Dozat, Timothy and Christopher D. Manning (2016). "Deep Biaffine Attention for Neural Dependency Parsing". In: *CoRR* abs/1611.01734. arXiv: 1611.01734. URL: http://arxiv.org/abs/1611.01734.

Dozat, Timothy and Christopher D Manning (2017). "Deep biaffine attention for neural dependency parsing". In: URL: https://openreview.net/forum?id=Hk95PK9le.

Dyer, Chris, Victor Chahuneau, and Noah A. Smith (June 2013). "A Simple, Fast, and Effective Reparameterization of IBM Model 2". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, pp. 644–648. URL: https://aclanthology.org/N13-1073.

Eisele, Andreas and Yu Chen (May 2010). "MultiUN: A Multilingual Corpus from United Nation Documents". In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA). URL: http://www.lrec-conf.org/proceedings/lrec2010/pdf/686_Paper.pdf.

Eisenschlos, Julian, Sebastian Ruder, Piotr Czapla, Marcin Kadras, Sylvain Gugger, and Jeremy Howard (Nov. 2019). "MultiFiT: Efficient Multi-lingual Language Model Fine-tuning". 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 5702–5707. URL: https://aclanthology.org/D19-1572.

El-Khair, Ibrahim Abu (2016). "1.5 billion words arabic corpus". In: *arXiv preprint arXiv:1611.04033*.

Fedus, William, Barret Zoph, and Noam Shazeer (2021). "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity". In: *arXiv preprint arXiv:2101.03961*.

BIBLIOGRAPHY

Fei, Hao, Meishan Zhang, and Donghong Ji (July 2020). "Cross-Lingual Semantic Role Labeling with High-Quality Translated Training Corpus". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 7014–7026. URL: https://aclanthology.org/2020.acl-main.627.

Gardner, Matt, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer (July 2018). "AllenNLP: A Deep Semantic Natural Language Processing Platform". In: *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*. Melbourne, Australia: Association for Computational Linguistics, pp. 1–6. URL: https://aclanthology.org/W18-2501.

Goodfellow, Ian J, Oriol Vinyals, and Andrew M Saxe (2014). "Qualitatively characterizing neural network optimization problems". In: *arXiv preprint arXiv:1412.6544*.

Goyal, Naman, Jingfei Du, Myle Ott, Giri Anantharaman, and Alexis Conneau (Aug. 2021). "Larger-Scale Transformers for Multilingual Masked Language Modeling". In: *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*. Online: Association for Computational Linguistics, pp. 29–33. URL: https://aclanthology.org/2021.repl4nlp-1.4.

He, Kaiming, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick (2020). "Momentum contrast for unsupervised visual representation learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738.

BIBLIOGRAPHY

Heinzerling, Benjamin and Michael Strube (May 2018). "BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://aclanthology.org/L18-1473.

Heinzerling, Benjamin and Michael Strube (July 2019). "Sequence Tagging with Contextual and Non-Contextual Subword Representations: A Multilingual Evaluation". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 273–291. URL: https://aclanthology.org/P19-1027.

Hendrycks, Dan and Kevin Gimpel (2016). "Bridging nonlinearities and stochastic regularizers with gaussian error linear units". In: *arXiv preprint arXiv:1606.08415*.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean (2015). "Distilling the knowledge in a neural network". In: *arXiv preprint arXiv:1503.02531*.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.

Howard, Jeremy and Sebastian Ruder (July 2018). "Universal Language Model Fine-tuning for Text Classification". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 328–339. URL: https://aclanthology.org/P18-1031.

BIBLIOGRAPHY

Hu, Junjie, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson (2020). "XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalisation". In: pp. 4411–4421.

Huang, Haoyang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou (Nov. 2019). "Unicoder: A Universal Language Encoder by Pre-training with Multiple Cross-lingual Tasks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 2485–2494. URL: https://aclanthology.org/D19-1252.

Jacobs, Robert A, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton (1991). "Adaptive mixtures of local experts". In: *Neural computation* 3.1, pp. 79–87.

Jalili Sabet, Masoud, Philipp Dufter, François Yvon, and Hinrich Schütze (Nov. 2020). "SimAlign: High Quality Word Alignments Without Parallel Training Data Using Static and Contextualized Embeddings". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 1627–1643. URL: https://aclanthology.org/2020.findings-emnlp.147.

Ji, Baijun, Zhirui Zhang, Xiangyu Duan, Min Zhang, Boxing Chen, and Weihua Luo (2020). "Cross-lingual pre-training based transfer for zero-shot neural machine translation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 01, pp. 115–122.

BIBLIOGRAPHY

Jordan, Michael I and Robert A Jacobs (1994). “Hierarchical mixtures of experts and the EM algorithm”. In: *Neural computation* 6.2, pp. 181–214.

K, Karthikeyan, Zihan Wang, Stephen Mayhew, and Dan Roth (2020). *Cross-Lingual Ability of Multilingual BERT: An Empirical Study*. URL: https://openreview.net/forum?id=HJeT3yrtDr.

Kale, Mihir, Aditya Siddhant, Rami Al-Rfou, Linting Xue, Noah Constant, and Melvin Johnson (Aug. 2021). “nmT5 - Is parallel data still relevant for pre-training massively 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 2: Short Papers)*. Online: Association for Computational Linguistics, pp. 683–691. URL: https://aclanthology.org/2021.acl-short.87.

Kamholz, David, Jonathan Pool, and Susan Colowick (May 2014). “PanLex: Building a Resource for Panlingual Lexical Translation”. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), pp. 3145–3150. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/1029_Paper.pdf.

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei (2020). “Scaling laws for neural language models”. In: *arXiv preprint arXiv:2001.08361*.

BIBLIOGRAPHY

Keung, Phillip, Yichao Lu, Julian Salazar, and Vikas Bhardwaj (Nov. 2020). "Don't Use English Dev: On the Zero-Shot Cross-Lingual Evaluation of Contextual Embeddings". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 549–554. URL: https://aclanthology.org/2020.emnlp-main.40.

Kim, Joo-Kyung, Young-Bum Kim, Ruhi Sarikaya, and Eric Fosler-Lussier (Sept. 2017). "Cross-Lingual Transfer Learning for POS Tagging without Cross-Lingual Resources". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 2832–2838. URL: https://aclanthology.org/D17-1302.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Kirkpatrick, James, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. (2017). "Overcoming catastrophic forgetting in neural networks". In: *Proceedings of the national academy of sciences* 114.13, pp. 3521–3526.

Koehn, Philipp (2005). "Europarl: A parallel corpus for statistical machine translation". In: *MT summit*. Vol. 5. Citeseer, pp. 79–86.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst (June 2007). "Moses: Open Source

Toolkit for Statistical Machine Translation". In: *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. Prague, Czech Republic: Association for Computational Linguistics, pp. 177–180. URL: https://aclanthology.org/P07-2045.

Kondratyuk, Dan and Milan Straka (Nov. 2019). "75 Languages, 1 Model: Parsing Universal Dependencies Universally". 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 2779–2795. URL: https://aclanthology.org/D19-1279.

Kornblith, Simon, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton (2019). "Similarity of neural network representations revisited". In: *International Conference on Machine Learning*.

Kudo, Taku (July 2018). "Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 66–75. URL: https://aclanthology.org/P18-1007.

Kudo, Taku and John Richardson (Nov. 2018). "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing:*

System Demonstrations. Brussels, Belgium: Association for Computational Linguistics, pp. 66–71. URL: https://aclanthology.org/D18-2012.

Kudugunta, Sneha, Ankur Bapna, Isaac Caswell, and Orhan Firat (Nov. 2019). "Investigating Multilingual NMT Representations at Scale". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 1565–1575. URL: https://aclanthology.org/D19-1167.

Kunchukuttan, Anoop, Pratik Mehta, and Pushpak Bhattacharyya (May 2018). "The IIT Bombay English-Hindi Parallel Corpus". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://aclanthology.org/L18-1548.

Laakso, Aarre and Garrison Cottrell (2000). "Content and cluster analysis: assessing representational similarity in neural systems". In: *Philosophical psychology* 13.1, pp. 47–76.

Lample, Guillaume, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou (2018). "Word translation without parallel data". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=H196sainb.

Lan, Wuwei, Yang Chen, Wei Xu, and Alan Ritter (Nov. 2020). “An Empirical Study of Pre-trained Transformers for Arabic Information Extraction”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 4727–4734. URL: https://aclanthology.org/2020.emnlp-main.382.

Lauscher, Anne, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš (Nov. 2020). “From Zero to Hero: On the Limitations of Zero-Shot Language Transfer with Multilingual Transformers”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 4483–4499. URL: https://aclanthology.org/2020.emnlp-main.363.

Lee, Cheolhyoung, Kyunghyun Cho, and Wanmo Kang (2020). “Mixout: Effective Regularization to Finetune Large-scale Pretrained Language Models”. In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=HkgaETNtDB.

Lepikhin, Dmitry, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen (2021). “{GS}hard: Scaling Giant Models with Conditional Computation and Automatic Sharding”. In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=qrwe7XHTmYb.

BIBLIOGRAPHY

Lester, Brian, Rami Al-Rfou, and Noah Constant (Nov. 2021). "The Power of Scale for Parameter-Efficient Prompt Tuning". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 3045–3059. URL: https://aclanthology.org/2021.emnlp-main.243.

Levow, Gina-Anne (July 2006). "The Third International Chinese Language Processing Bakeoff: Word Segmentation and Named Entity Recognition". In: *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*. Sydney, Australia: Association for Computational Linguistics, pp. 108–117. URL: https://aclanthology.org/W06-0115.

Li, Hao, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein (2018). "Visualizing the Loss Landscape of Neural Nets". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2018/file/a41b3bb3e6b050b6c9067c67f663b915-Paper.pdf.

Li, Xiang Lisa and Percy Liang (Aug. 2021). "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)*. Online: Association for Computational

Linguistics, pp. 4582–4597. URL: https://aclanthology.org/2021.acl-long.353.

Li, Xuansong, Stephanie Strassel, Stephen Grimes, Safa Ismael, Mohamed Maamouri, Ann Bies, and Nianwen Xue (May 2012). "Parallel Aligned Treebanks at LDC: New Challenges Interfacing Existing Infrastructures". In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA), pp. 1848–1855. URL: http://www.lrec-conf.org/proceedings/lrec2012/pdf/277_Paper.pdf.

Li, Yixuan, Jason Yosinski, Jeff Clune, Hod Lipson, and John E Hopcroft (2016). "Convergent Learning: Do different neural networks learn the same representations?" In: *Iclr*.

Liang, Yaobo, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, et al. (2020). "XGLUE: A New Benchmark Dataset for Cross-lingual Pre-training, Understanding and Generation". In: *arXiv preprint arXiv:2004.01401*.

Lin, Ying, Heng Ji, Fei Huang, and Lingfei Wu (July 2020). "A Joint Neural Model for Information Extraction with Global Features". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 7999–8009. URL: https://aclanthology.org/2020.acl-main.713.

BIBLIOGRAPHY

Lison, Pierre, Jörg Tiedemann, and Milen Kouylekov (May 2018). "OpenSubtitles2018: Statistical Rescoring of Sentence Alignments in Large, Noisy Parallel Corpora". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://aclanthology.org/L18-1275.

Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig (2021a). "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing". In: *arXiv preprint arXiv:2107.13586*.

Liu, Qianchu, Diana McCarthy, Ivan Vulić, and Anna Korhonen (Nov. 2019a). "Investigating Cross-Lingual Alignment Methods for Contextualized Embeddings with Token-Level Evaluation". In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics, pp. 33–43. URL: https://aclanthology.org/K19-1004.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019b). "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *arXiv preprint arXiv:1907.11692*.

Liu, Zihan, Genta Indra Winata, Andrea Madotto, and Pascale Fung (Aug. 2021b). "Preserving Cross-Linguality of Pre-trained Models via Continual Learning". In: *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*. Online: Association for Computational Linguistics, pp. 64–71. URL: https://aclanthology.org/2021.repl4nlp-1.8.

BIBLIOGRAPHY

Martin, Louis, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot (July 2020). "CamemBERT: a Tasty French Language Model". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 7203–7219. URL: https://aclanthology.org/2020.acl-main.645.

McCann, Bryan, James Bradbury, Caiming Xiong, and Richard Socher (2017). "Learned in Translation: Contextualized Word Vectors". In: *Advances in Neural Information Processing Systems* 30.

McCloskey, Michael and Neal J. Cohen (1989). "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem". In: *Psychology of Learning and Motivation* 24, pp. 109–165.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013a). "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781*.

Mikolov, Tomas, Quoc V Le, and Ilya Sutskever (2013). "Exploiting similarities among languages for machine translation". In: *arXiv preprint arXiv:1309.4168*.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013b). "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*, pp. 3111–3119.

Morcos, Ari, Maithra Raghu, and Samy Bengio (2018). “Insights on representational similarity in neural networks with canonical correlation”. In: *Advances in Neural Information Processing Systems*, pp. 5727–5736.

Mosbach, Marius, Maksym Andriushchenko, and Dietrich Klakow (2021). “On the Stability of Fine-tuning BERT: Misconceptions, Explanations, and Strong Baselines”. In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=nzpLWnVAyah.

Mulcaire, Phoebe, Jungo Kasai, and Noah A. Smith (June 2019). “Polyglot Contextual Representations Improve Crosslingual Transfer”. 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 3912–3918. URL: https://aclanthology.org/N19-1392.

Nguyen, Dat Quoc and Anh Tuan Nguyen (Nov. 2020). “PhoBERT: Pre-trained language models for Vietnamese”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 1037–1042. URL: https://aclanthology.org/2020.findings-emnlp.92.

Nivre, Joakim et al. (2016). *Universal Dependencies 1.4*. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. URL: http://hdl.handle.net/11234/1-1827.

Nivre, Joakim et al. (2018a). *Universal Dependencies 2.2*. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. URL: http://hdl.handle.net/11234/1-2837.

Nivre, Joakim et al. (2018b). *Universal Dependencies 2.3*. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. URL: http://hdl.handle.net/11234/1-2895.

Och, Franz Josef and Hermann Ney (2003). "A Systematic Comparison of Various Statistical Alignment Models". In: *Computational Linguistics* 29.1, pp. 19–51. URL: https://aclanthology.org/J03-1002.

Ormazabal, Aitor, Mikel Artetxe, Gorka Labaka, Aitor Soroa, and Eneko Agirre (July 2019). "Analyzing the Limitations of Cross-lingual Word Embedding Mappings". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 4990–4995. URL: https://aclanthology.org/P19-1492.

Ortiz Suárez, Pedro Javier, Laurent Romary, and Benoît Sagot (July 2020). "A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 1703–1714. URL: https://aclanthology.org/2020.acl-main.156.

Ott, Myle, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli (June 2019). "fairseq: A Fast, Extensible Toolkit for Se-

quence Modeling". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 48–53. URL: https://aclanthology.org/N19-4009.

Ozaki, Hiroaki, Gaku Morio, Terufumi Morishita, and Toshinori Miyoshi (Apr. 2021). "Project-then-Transfer: Effective Two-stage Cross-lingual Transfer for Semantic Dependency Parsing". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, pp. 2586–2594. URL: https://aclanthology.org/2021.eacl-main.221.

Pan, Sinno Jialin and Qiang Yang (2010). "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10, pp. 1345–1359.

Pan, Xiaoman, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji (July 2017). "Cross-lingual Name Tagging and Linking for 282 Languages". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 1946–1958. URL: https://aclanthology.org/P17-1178.

Panchendrarajan, Rrubaa and Aravindh Amaresan (2018). "Bidirectional LSTM-CRF for Named Entity Recognition". In: *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*. Hong Kong: Association for Computational Linguistics. URL: https://aclanthology.org/Y18-1061.

Parker, Robert, David Graff, Ke Chen, Junbo Kong, and Kazuaki Maeda (2011). "Arabic Gigaword fifth edition LDC2011T11". In: *Philadelphia: Linguistic Data Consortium*.

Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. (2019). "PyTorch: An imperative style, high-performance deep learning library". In: *Advances in Neural Information Processing Systems*, pp. 8024–8035.

Patra, Barun, Joel Ruben Antony Moniz, Sarthak Garg, Matthew R. Gormley, and Graham Neubig (July 2019). "Bilingual Lexicon Induction with Semi-supervision in Non-Isometric Embedding Spaces". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 184–193. URL: https://aclanthology.org/P19-1018.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (Oct. 2014). "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. URL: https://aclanthology.org/D14-1162.

Perkins, Jacob (2014). *Python 3 text processing with NLTK 3 cookbook*. Packt Publishing Ltd.

Peters, Matthew E., Waleed Ammar, Chandra Bhagavatula, and Russell Power (July 2017). "Semi-supervised sequence tagging with bidirectional language models". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1:*

Long Papers). Vancouver, Canada: Association for Computational Linguistics, pp. 1756–1765. URL: https://aclanthology.org/P17-1161.

Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (June 2018). "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 2227–2237. URL: https://aclanthology.org/N18-1202.

Pfeiffer, Jonas, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder (2020). "Unks everywhere: Adapting multilingual language models to new scripts". In: *arXiv preprint arXiv:2012.15562*.

Phang, Jason, Thibault Févry, and Samuel R Bowman (2018). "Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks". In: *arXiv preprint arXiv:1811.01088*.

Pires, Telmo, Eva Schlinger, and Dan Garrette (July 2019). "How Multilingual is Multilingual BERT?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 4996–5001. URL: https://aclanthology.org/P19-1493.

Post, Matt (Oct. 2018). "A Call for Clarity in Reporting BLEU Scores". In: *Proceedings of the Third Conference on Machine Translation: Research Papers*. Brussels,

Belgium: Association for Computational Linguistics, pp. 186–191. URL: https://aclanthology.org/W18-6319.

Press, Ofir and Lior Wolf (Apr. 2017). "Using the Output Embedding to Improve Language Models". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 157–163. URL: https://aclanthology.org/E17-2025.

Qin, Guanghui and Jason Eisner (June 2021). "Learning How to Ask: Querying LMs with Mixtures of Soft Prompts". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 5203–5212. URL: https://aclanthology.org/2021.naacl-main.410.

Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever (2018). "Improving language understanding by generative pre-training". In.

Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). "Language Models are Unsupervised Multitask Learners". In.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu (2020). "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *Journal of Machine Learning Research* 21.140, pp. 1–67.

Raghu, Maithra, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein (2017). "Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability". In: *Advances in Neural Information Processing Systems*, pp. 6076–6085.

Řehůřek, Radim and Petr Sojka (May 2010). "Software Framework for Topic Modelling with Large Corpora". English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. http://is.muni.cz/publication/884893/en. Valletta, Malta: ELRA, pp. 45–50.

Riloff, Ellen, Charles Schafer, and David Yarowsky (2002). "Inducing Information Extraction Systems for New Languages via Cross-language Projection". In: *COLING 2002: The 19th International Conference on Computational Linguistics*. URL: https://aclanthology.org/C02-1070.

Ruder, Sebastian, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Dan Garrette, Graham Neubig, and Melvin Johnson (Nov. 2021). "XTREME-R: Towards More Challenging and Nuanced Multilingual Evaluation". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 10215–10245. URL: https://aclanthology.org/2021.emnlp-main.802.

Ruder, Sebastian, Ivan Vulić, and Anders Søgaard (2019). "A survey of cross-lingual word embedding models". In: *Journal of Artificial Intelligence Research* 65, pp. 569–631.

Schuster, Tal, Ori Ram, Regina Barzilay, and Amir Globerson (June 2019). "Cross-Lingual Alignment of Contextual Word Embeddings, with Applications to Zero-shot Dependency Parsing". 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 1599–1613. URL: https://aclanthology.org/N19-1162.

Schwenk, Holger, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán (Apr. 2021). "WikiMatrix: Mining 135M Parallel Sentences in 1620 Language Pairs from Wikipedia". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, pp. 1351–1361. URL: https://aclanthology.org/2021.eacl-main.115.

Schwenk, Holger and Xian Li (May 2018). "A Corpus for Multilingual Document Classification in Eight Languages". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://aclanthology.org/L18-1560.

Schwenk, Holger, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin (2019). "CCMatrix: Mining Billions of High-Quality Parallel Sentences on the WEB". In: *arXiv preprint arXiv:1911.04944*.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (Aug. 2016). "Neural Machine Translation of Rare Words with Subword Units". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1715–1725. URL: https://aclanthology.org/P16-1162.

Shazeer, Noam, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean (2017). "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer". In: *arXiv preprint arXiv:1701.06538*.

Skadiņš, Raivis, Jörg Tiedemann, Roberts Rozis, and Daiga Deksne (May 2014). "Billions of Parallel Words for Free: Building and Using the EU Bookshop Corpus". In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), pp. 1850–1855. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/846_Paper.pdf.

Smith, Samuel L, David HP Turban, Steven Hamblin, and Nils Y Hammerla (2017). "Offline bilingual word vectors, orthogonal transformations and the inverted softmax". In: *arXiv preprint arXiv:1702.03859*.

Søgaard, Anders, Sebastian Ruder, and Ivan Vulić (July 2018). "On the Limitations of Unsupervised Bilingual Dictionary Induction". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Melbourne, Australia: Association for Computational Linguistics, pp. 778–788. URL: https://aclanthology.org/P18-1072.

Stengel-Eskin, Elias, Tzu-ray Su, Matt Post, and Benjamin Van Durme (Nov. 2019). "A Discriminative Neural Model for Cross-Lingual Word Alignment". 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 910–920. URL: https://aclanthology.org/D19-1084.

Taylor, Wilson L (1953). ""Cloze procedure": A new tool for measuring readability". In: *Journalism Bulletin* 30.4, pp. 415–433.

Thoma, Martin (2018). "The WiLI benchmark dataset for written language identification". In: *arXiv preprint arXiv:1801.07779*.

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

Tiedemann, Jörg (Nov. 2020). "The Tatoeba Translation Challenge – Realistic Data Sets for Low Resource and Multilingual MT". In: *Proceedings of the Fifth Conference on Machine Translation*. Online: Association for Computational Linguistics, pp. 1174–1182. URL: https://aclanthology.org/2020.wmt-1.139.

Tiedemann, Jörg, Željko Agić, and Joakim Nivre (June 2014). "Treebank Translation for Cross-Lingual Parser Induction". In: *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 130–140. URL: https://aclanthology.org/W14-1614.

Tjong Kim Sang, Erik F. (2002). "Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition". In: *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*. URL: https://aclanthology.org/W02-2024.

Tjong Kim Sang, Erik F. and Fien De Meulder (2003). "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition". In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 142–147. URL: https://aclanthology.org/W03-0419.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: *Advances in Neural Information Processing Systems*, pp. 5998–6008.

Walker, Christopher, Stephanie Strassel, Julie Medero, and Kazuaki Maeda (2006). "ACE 2005 Multilingual Training Corpus LDC2006T06". In: *Web Download. Philadelphia: Linguistic Data Consortium*.

Wang, Liwei, Lunjia Hu, Jiayuan Gu, Zhiqiang Hu, Yue Wu, Kun He, and John Hopcroft (2018). "Towards understanding learning representations: To what extent do different

neural networks learn the same representation". In: *Advances in Neural Information Processing Systems*, pp. 9584–9593.

Wang, Yuxuan, Wanxiang Che, Jiang Guo, Yijia Liu, and Ting Liu (Nov. 2019). "Cross-Lingual BERT Transformation for Zero-Shot Dependency Parsing". 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 5721–5727. URL: https://aclanthology.org/D19-1575.

Wang, Zihan, Karthikeyan K, Stephen Mayhew, and Dan Roth (Nov. 2020a). "Extending Multilingual BERT to Low-Resource Languages". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 2649–2656. URL: https://aclanthology.org/2020.findings-emnlp.240.

Wang, Zirui, Jiateng Xie, Ruochen Xu, Yiming Yang, Graham Neubig, and Jaime G. Carbonell (2020b). "Cross-lingual Alignment vs Joint Training: A Comparative Study and A Simple Unified Framework". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=S1l-C0NtwS.

Winata, Genta Indra, Andrea Madotto, Zhaojiang Lin, Rosanne Liu, Jason Yosinski, and Pascale Fung (2021). "Language Models are Few-shot Multilingual Learners". In: *arXiv preprint arXiv:2109.07684*.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush (Oct. 2020). "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, pp. 38–45. URL: https://aclanthology.org/2020.emnlp-demos.6.

Wu, Shijie and Mark Dredze (Nov. 2019). "Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 833–844. URL: https://aclanthology.org/D19-1077.

Wu, Shijie and Mark Dredze (July 2020a). "Are All Languages Created Equal in Multilingual BERT?" In: *Proceedings of the 5th Workshop on Representation Learning for NLP*. Online: Association for Computational Linguistics, pp. 120–130. URL: https://aclanthology.org/2020.repl4nlp-1.16.

Wu, Shijie and Mark Dredze (Nov. 2020b). "Do Explicit Alignments Robustly Improve Multilingual Encoders?" In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational

Linguistics, pp. 4471–4482. URL: https://aclanthology.org/2020.emnlp-main.362.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. (2016). "Google's neural machine translation system: Bridging the gap between human and machine translation". In: *arXiv preprint arXiv:1609.08144*.

Xia, Patrick, Guanghui Qin, Siddharth Vashishtha, Yunmo Chen, Tongfei Chen, Chandler May, Craig Harman, Kyle Rawlins, Aaron Steven White, and Benjamin Van Durme (Apr. 2021). "LOME: Large Ontology Multilingual Extraction". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. Online: Association for Computational Linguistics, pp. 149–159. URL: https://aclanthology.org/2021.eacl-demos.19.

Xie, Jiateng, Zhilin Yang, Graham Neubig, Noah A. Smith, and Jaime Carbonell (2018). "Neural Cross-Lingual Named Entity Recognition with Minimal Resources". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 369–379. URL: https://aclanthology.org/D18-1034.

Xu, Haoran, Seth Ebner, Mahsa Yarmohammadi, Aaron Steven White, Benjamin Van Durme, and Kenton Murray (Apr. 2021). "Gradual Fine-Tuning for Low-Resource Domain Adaptation". In: *Proceedings of the Second Workshop on Domain Adaptation*

for NLP. Kyiv, Ukraine: Association for Computational Linguistics, pp. 214–221. URL: https://aclanthology.org/2021.adaptnlp-1.22.

Xu, Haoran, Benjamin Van Durme, and Kenton Murray (Nov. 2021). "BERT, mBERT, or BiBERT? A Study on Contextualized Embeddings for Neural Machine Translation". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 6663–6675. URL: https://aclanthology.org/2021.emnlp-main.534.

Xue, Linting, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel (June 2021). "mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 483–498. URL: https://aclanthology.org/2021.naacl-main.41.

Yarmohammadi, Mahsa, Shijie Wu, Marc Marone, Haoran Xu, Seth Ebner, Guanghui Qin, Yunmo Chen, Jialiang Guo, Craig Harman, Kenton Murray, Aaron Steven White, Mark Dredze, and Benjamin Van Durme (Nov. 2021). "Everything Is All It Takes: A Multipronged Strategy for Zero-Shot Cross-Lingual Information Extraction". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational

Linguistics, pp. 1950–1967. URL: https://aclanthology.org/2021.emnlp-main.149.

Yarowsky, David (June 1995). "Unsupervised Word Sense Disambiguation Rivaling Supervised Methods". In: *33rd Annual Meeting of the Association for Computational Linguistics*. Cambridge, Massachusetts, USA: Association for Computational Linguistics, pp. 189–196. URL: https://aclanthology.org/P95-1026.

Yarowsky, David and Grace Ngai (2001). "Inducing Multilingual POS Taggers and NP Bracketers via Robust Projection Across Aligned Corpora". In: *Second Meeting of the North American Chapter of the Association for Computational Linguistics*. URL: https://aclanthology.org/N01-1026.

Yarowsky, David, Grace Ngai, and Richard Wicentowski (2001). "Inducing Multilingual Text Analysis Tools via Robust Projection across Aligned Corpora". In: *Proceedings of the First International Conference on Human Language Technology Research*. URL: https://aclanthology.org/H01-1035.

Yosinski, Jason, Jeff Clune, Yoshua Bengio, and Hod Lipson (2014). "How transferable are features in deep neural networks?" In: *Advances in neural information processing systems*, pp. 3320–3328.

Zeman, Daniel and Philip Resnik (2008). "Cross-Language Parser Adaptation between Related Languages". In: *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*. URL: https://aclanthology.org/I08-3008.

Zeman, Daniel et al. (2020a). *Universal Dependencies 2.6*. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. URL: http://hdl.handle.net/11234/1-3226.

Zeman, Daniel et al. (2020b). *Universal Dependencies 2.7*. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. URL: http://hdl.handle.net/11234/1-3424.

Zeroual, Imad, Dirk Goldhahn, Thomas Eckart, and Abdelhak Lakhouaja (Aug. 2019). "OSIAN: Open Source International Arabic News Corpus - Preparation and Integration into the CLARIN-infrastructure". In: *Proceedings of the Fourth Arabic Natural Language Processing Workshop*. Florence, Italy: Association for Computational Linguistics, pp. 175–182. URL: https://aclanthology.org/W19-4619.

Zhang, Biao, Philip Williams, Ivan Titov, and Rico Sennrich (July 2020). "Improving Massively Multilingual Neural Machine Translation and Zero-Shot Translation". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 1628–1639. URL: https://aclanthology.org/2020.acl-main.148.

Zhang, Meng, Yang Liu, Huanbo Luan, and Maosong Sun (July 2017). "Adversarial Training for Unsupervised Bilingual Lexicon Induction". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Vancouver, Canada: Association for Computational Linguistics, pp. 1959–1970. URL: https://aclanthology.org/P17-1179.

Zhang, Mozhi, Keyulu Xu, Ken-ichi Kawarabayashi, Stefanie Jegelka, and Jordan Boyd-Graber (July 2019). "Are Girls Neko or Shōjo? Cross-Lingual Alignment of Non-Isomorphic Embeddings with Iterative Normalization". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 3180–3189. URL: https://aclanthology.org/P19-1307.

Zhang, Tianyi, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi (2021). "Revisiting Few-sample BERT Fine-tuning". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=cO1IH43yUF.

Zhao, Mengjie, Yi Zhu, Ehsan Shareghi, Ivan Vulić, Roi Reichart, Anna Korhonen, and Hinrich Schütze (Aug. 2021). "A Closer Look at Few-Shot Crosslingual Transfer: The Choice of Shots Matters". 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)*. Online: Association for Computational Linguistics, pp. 5751–5767. URL: https://aclanthology.org/2021.acl-long.447.

Vita

Shijie Wu received a B.S. degree in Informational and Computational Science from Sun Yat-sen University in 2016 and a M.S.E. degree in Computer Science from Johns Hopkins University in 2018. Shijie enrolled in the Ph.D. program in Computer Science at Johns Hopkins University in 2018. Shijie's research focuses on cross-lingual transfer with pretrained multilingual encoders, ranging from discovering its cross-lingual potential, understanding why it works, documenting its caveat, to figuring out ways to improve it. Outside of the main area, Shijie also worked on papers related to morphology and computational linguistics.